3GPP2 C.S0052-0

Version 1.0

Date: June 11, 2004

Source-Controlled Variable-Rate Multimode Wideband Speech Codec (VMR-WB)

Service Option 62 for Spread Spectrum Systems

Intentionally left blank.

FOREWORD

These technical requirements form a standard for Service Option 62, source-controlled variable-rate multimode two-way wideband speech Service Option (VMR-WB). VMR-WB has a number of operating modes where each mode corresponds to a certain quality and average data rate and all modes are fully compliant with Rate-Set II of CDMA systems. The maximum speech-coding rate of the Service Option 62 is 13.3 kbps.

VMR-WB standard is also interoperable with 3GPP/AMR-WB (ITU-T/G.722.2) standard at 12.65, 8.85, and 6.60 kbps. The VMR-WB acronym has been chosen to reflect the algorithmic similarities and interoperability between the two codecs. This document further describes the necessary interworking functions for establishing an interoperable interconnection between VMR-WB and AMR-WB. The applications of the AMR-WB interoperable mode and methods for initiation and setup of interoperable calls are beyond the scope of this specification.

The VMR-WB standard, while a wideband speech codec by default, is capable of processing narrowband input speech signals and produce narrowband outputs in all modes of operation. Therefore, this document further describes procedures for initialization and call setup using narrowband speech processing capability of VMR-WB codec.

This standard does not address the quality or reliability of Service Option 62, nor does it cover equipment performance or measurement procedures.

NOTES

1. The associated 3GPP2 C.S0053-0, “Recommended Minimum Performance Standard for the variable-rate multimode wideband speech codec, Service Option 62,” provides specifications and measurement methods.

2. “Base station” refers to the functions performed on the landline side, which are typically distributed among a cell, a sector of a cell, a mobile switching center, and a personal communications switching center.

3. This document uses the following verbal forms: “Shall” and “shall not” identify requirements to be followed strictly to conform to the standard and from which no deviation is permitted. “Should” and “should not” indicate that one of several possibilities is recommended as particularly suitable, without mentioning or excluding others; that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. “May” and “need not” indicate a course of action permissible within the limits of the standard. “Can” and “cannot” are used for statements of possibility and capability, whether material, physical, or causal.

4. Footnotes appear at various points in this specification to elaborate and further clarify items discussed in the body of the specification.

5. Unless indicated otherwise, this document presents numbers in decimal form.

   Binary numbers are distinguished in the text by the use of single quotation marks. In some tables, binary values may appear without single quotation marks if table notation clearly specifies that values are binary. The character ‘x’ is used to represent a binary bit of unspecified value. For example ‘xxx00010’ represents any 8-bit binary value such that the least significant five bits equal ‘00010’.

   Hexadecimal numbers (base 16) are distinguished in the text by use of the form 0xh…h, where h…h represents a string of hexadecimal digits. For example, 0x2FA1 represents a number whose binary value is ‘10111110100001’ and whose decimal value is 12913.

NOTES

6. The following conventions apply to mathematical expressions in this standard:

- $\lfloor x \rfloor$ indicates the largest integer less than or equal to x: $\lfloor 1.1 \rfloor$= 1, $\lfloor 1.0 \rfloor$= 1, and $\lfloor -1.1 \rfloor$=-2.
- $\lceil x \rceil$ indicates the smallest integer greater than or equal to x: $\lceil 1.1 \rceil$ = 2, $\lceil 2.0 \rceil$ = 2, and $\lceil -1.1 \rceil$=-1.
- $|x|$ indicates the absolute value of x: $|-17|$=17, $|17|$=17.
- $\oplus$ indicates exclusive OR.
- min(x, y) indicates the minimum of x and y.
- max(x, y) indicates the maximum of x and y.
- In figures, $\otimes$ indicates multiplication. In formulas within the text, multiplication is implicit. For example, if h(n) and $p_L(n)$ are functions, then h(n) $p_L(n)$ = h(n) $\otimes$ $p_L(n)$.
- x mod y indicates the remainder after dividing x by y: x mod y = x - (y$\lfloor x / y \rfloor$).
- round(x) is traditional rounding: round(x) = sign(x)$\lfloor |x| + 0.5 \rfloor$, where

$$\mathrm{sign}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}.$$

- $\sum$ indicates summation. If the summation symbol specifies initial and terminal values, and the initial value is greater than the terminal value, then the value of the summation is 0. For example, if N=0, and if f(n) represents an arbitrary function, then

$$\sum_{n=1}^{N} f(n) = 0.$$

- The bracket operator, [ ], isolates individual bits of a binary value. VAR[n] refers to bit n of the binary representation of the value of the variable VAR, such that VAR[0] is the least significant bit of VAR. The value of VAR[n] is either 0 or 1.
- Unless otherwise specified log(x) denotes logarithm at base 10 throughout this document.

REFERENCES

The following standards contain provisions; through reference in this text constitute provisions of this Standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this Standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. ANSI and TIA maintain registers of currently valid national standards published by them.

—*Normative References:*

1. ANSI/EIA/TIA-579-A-98, *Telecommunications Telephone Terminal Equipment Transmission Requirements for Digital Wireline Telephones*, Nov. 1998.
2. ITU-T Recommendation G.711, *Pulse Code Modulation (PCM) of Voice Frequencies*, Vol. III, Geneva, November 1988.
3. ITU-T Recommendation G.712, *Transmission Performance Characteristics of Pulse Code Modulation,* November 2001.
4. ITU-T Recommendation P.79, *Calculation of loudness ratings for telephone sets*, September 1999.
5. IEEE Standard 269-2002, *Standard Method for Measuring Transmission Performance of Analog and Digital Telephone Set, Handsets and Headset,* 2002.
6. 3GPP2 C.S0003-0 v3.0, *Medium Access Control (MAC) Standard for cdma2000 Spread Spectrum Systems,* July 2001.
7. 3GPP2 C.S0005-0 v3.0, *Upper Layer (Layer 3) Signaling Standard for cdma2000 Spread Spectrum Systems,* July 2001.
8. 3GPP2 C.S0014-0, *Enhanced Variable Rate Codec, Speech Service Option 3 for Wideband Spread Spectrum Digital Systems*, December 1999.
9. 3GPP TS 26.190, *AMR Wideband Speech Codec; Transcoding functions,* December 2001.
10. 3GPP TS 26.173, *AMR Wideband Speech Codec; ANSI-C code,* March 2002.
11. 3GPP TS 26.174, *AMR Wideband Speech Codec; Test seq*uences*,* December 2001.
12. 3GPP TS 26.193, *AMR Wideband Speech Codec; Source Controlled Rate Operation,* March 2001.
13. 3GPP TS 26.194, *AMR Wideband Speech Codec; Voice Activity Detection (VAD),* March 2001.
14. 3GPP TS 26.192, *AMR Wideband Speech Codec; Comfort Noise Aspects,* March 2001.
15. 3GPP TS 26.191, *AMR Wideband Speech Codec; Error Concealment of Lost Frames,* March 2002.
16. 3GPP TS 26.201, *AMR Wideband Speech Codec; Frame Structure,* March 2001.
17. 3GPP TS 26.202, *AMR Wideband Speech Codec; Interface to RAN,* March 2001.
18. 3GPP TS 26.976, *AMR Wideband Speech Codec; Performance characterisation,* June 2001.

—*Informative References:*

19. J. D. Johnston, "Transform Coding of Audio Signals Using Perceptual Noise Criteria," IEEE Journal on Selected Areas in Communications, vol. 6, no. 2, pp. 314-323, February 1988.

20. W.B. Kleijn, P. Kroon, and D. Nahumi, “The RCELP speech-coding algorithm,” European Transactions on Telecommunications, Vol. 4, No. 5, pp. 573–582, 1994.

21. L.R. Rabiner and R.W. Schaefer, *Digital processing of speech signals*, Prentice-Hall Int., 1978

22. Y. Bistritz and S. Pellerm “Immitance Spectral Pairs (ISP) for speech encoding”, in Proceedings of ICASSP'93, pp. II-9 – II-12, 1993.

23. P. Kabal and R.P. Ramachandran, “The computation of line spectral frequencies using Chebyshev polynomials”, *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol. 34, no. 6, pp. 1419-1426, 1986

24. ITU-T Recommendation P.341, *Transmission Characteristics for Wideband (150-7000 Hz) Digital Hands-Free Telephony Terminals, February 1998.*

25. ITU-T Recommendation P.311, *Transmission Characteristics for Wideband (150-7000 Hz) Digital Handset Telephones,* February 1998.

Table of Contents

1 Source-controlled Variable-rate multimode Two-Way wideband Voice Communication (VMR-WB) ... 12
1.1 General Description ... 12
1.2 Overview of VMR-WB Documentation ... 12
1.3 VMR-WB Implementation Options ... 13
1.4 VMR-WB Algorithmic Options ... 13
1.5 Service Option Number ... 14
1.6 Allowable Delays ... 14
1.6.1 Allowable Transmitting Speech Codec Encoding Delay ... 14
1.6.2 Allowable Receiving Speech Codec Decoding Delay ... 14
1.7 Special Cases ... 14
1.7.1 Blanked Packets ... 14
1.7.2 Null Traffic Channel Data ... 14
1.7.3 All Zeros Packet ... 14
1.8 Terms and Numeric Information ... 15
2 Required Multiplex Option Support ... 18
2.1 Interface to Multiplex Option 2 ... 18
2.1.1 Transmitted Packets ... 18
2.1.2 Received Packets ... 18
2.2 Negotiation for Service Option 62 ... 19
2.2.1 Procedures Using Service Negotiation ... 19
2.2.1.1 Initialization and Connection ... 19
2.2.1.1.1 Mobile Station Requirements ... 19
2.2.1.1.2 Base Station Requirements ... 20
2.2.1.2 Service Option Control Messages ... 20
2.2.1.2.1 Mobile Station Requirements ... 20
2.2.1.2.2 Base Station Requirements ... 21
3 Audio Interfaces ... 23
3.1 Input Audio Interface ... 23
3.1.1 Input Audio Interface in the Mobile Station ... 23
3.1.1.1 Conversion and Scaling ... 24
3.1.1.2 Digital Audio Input ... 24
3.1.1.3 Analog Audio Input ... 24
3.1.1.3.1 Transmit Level Adjustment ... 24
3.1.1.3.2 Band Pass Filtering ... 24
3.1.1.3.3 Echo Return Loss ... 24
3.1.2 Input Audio Interface in the Base Station ... 25
3.1.2.1 Sampling and Format Conversion ... 25
3.1.2.2 Transmit Level Adjust ... 25
3.1.2.3 Line Echo Canceling ... 25
3.2 Output Audio Interface ... 25
3.2.1 Output Audio Interface in the Mobile Station ... 25
3.2.1.1 Band Pass Filtering ... 25
3.2.1.2 Receive Level Adjustment ... 25
3.2.2 Output Audio Interface in the Base Station ... 25
3.2.2.1 Receive Level Adjustment ... 25
4 The variable-rate multimode wideband speech codec (VMR-WB) – Introduction and Bit Allocation Tables ... 26
4.1 Introduction to the VMR-WB Speech Coding Algorithm ... 26
4.2 Bit Allocation Tables ... 32
4.3 VMR-WB Symbol Table ... 34
4.4 Abbreviations ... 37
5 Functional description of the VMR-WB Encoder ... 39

5.1 Pre-Processing 39
5.1.1 Sampling Conversion 39
5.1.2 High-Pass Filtering and Pre-emphasis 41
5.2 Spectral Analysis 42
5.2.1 Detection of narrowband inputs 44
5.3 Voice Activity Detection 45
5.4 Primary Noise Parameter Estimation and Update 46
5.4.1 Total noise and Relative Frame Energy Estimation 47
5.4.2 Frame Energy per Critical Band, Noise Initialization, and Noise Update 47
5.4.3 Long-Term Average Noise Energy and Frame Energy Update 48
5.4.4 Noise Correction Factor 49
5.5 Noise Suppression 49
5.5.1 Reconstruction of Denoised Signal 53
5.6 Linear Prediction Analysis and ISP Conversion 54
5.6.1 Windowing and Autocorrelation Computation 54
5.6.2 Levinson-Durbin Algorithm 55
5.6.3 LP to ISP conversion 55
5.6.4 ISP to LP Conversion 57
5.6.5 Interpolation of ISPs 58
5.7 Perceptual Weighting 58
5.8 Open-loop Pitch Analysis and Pitch Tracking 60
5.8.1 Correlation Function Computation 60
5.8.2 Correlation Reinforcement with Past Pitch Values 61
5.8.3 Normalized Correlation Computation 62
5.8.4 Correlation Reinforcement with Pitch Lag Multiples 62
5.8.5 Initial Pitch Lag Determination and Reinforcement Based on Pitch Coherence with other Half-frames 63
5.8.6 Pitch Lag Determination and Parameter Update 63
5.9 Noise Energy Estimate Update and Voiced Critical Band Determination 64
5.9.1 Update of Voicing Cutoff Frequency 66
5.10 Unvoiced Signal Classification: Selection of Unvoiced-HR and Unvoiced-QR 67
5.10.1 Voicing Measure 69
5.10.2 Spectral Tilt 69
5.10.3 Energy Variation 71
5.10.4 Relative Frame Energy E_{rel} 71
5.10.5 Unvoiced Speech Classification 71
5.11 Signal Modification and HR Voiced Rate Selection 73
5.11.1 Search of Pitch Pulses and Pitch Cycle Segments 74
5.11.2 Determination of the Delay Parameter 78
5.11.3 Modification of the Signal 81
5.11.4 Voiced Classification Logic Incorporated into the Signal Modification Procedure 85
5.12 Selection of FR and Generic HR, and Maximum and Minimum Rate Operation 87
5.12.1 Maximum and Minimum Rate Operation 88
5.13 Quantization of the ISP Coefficients 89
5.14 Impulse Response Computation 92
5.15 Target Signal Computation 93
5.16 Adaptive Codebook Search 93
5.16.1 Adaptive Codebook Search in Full Rate Encoding Type 94
5.16.2 Adaptive Codebook Search in Generic HR 95
5.16.3 Computation of Adaptive Codebook Excitation in FR and Generic HR 95
5.16.4 Computation of Adaptive Codebook Excitation in Voiced HR 96
5.16.5 Frequency Dependent Pitch Prediction 96
5.16.6 Computation of Adaptive Codebook Gain 96
5.17 Algebraic Codebook for FR, Voiced HR, and Generic HR 97
5.17.1 Codebook Structure 98
5.17.1.1 FR Encoding Types 98

5.17.1.2 Voiced HR and Generic HR Encoding Types 99
5.17.2 Algebraic Codebook Search 100
5.17.2.1 Codebook Search in FR Encoding Types 101
5.17.2.2 Codebook Search in Voiced HR and Generic HR 104
5.18 Gaussian Codebook Structure and Search in Unvoiced HR 104
5.18.1 Structure of the random codebook 104
5.18.2 Search of the Random Codebook 105
5.19 Random Excitation in Unvoiced QR 107
5.20 Quantization of the Adaptive and Fixed-Codebook Gains 108
5.20.1 Gain Quantization in Unvoiced HR and Unvoiced QR 110
5.21 Memory Update 111
5.22 Supplementary Information for Frame Error Concealment in Generic FR 112
5.22.1 Signal Classification for Frame Error Concealment and Recovery 112
5.22.2 Other Speech Parameters for Frame Error Processing 116
5.22.2.1 Energy Information 116
5.22.2.2 Phase Control Information 117
5.23 Encoding of Inactive Speech Frames (CNG-ER and CNG-QR) 117
5.23.1 LP Parameter Quantization in CNG-ER and CNG-QR 118
5.23.2 Energy Quantization in CNG-ER and CNG-QR 119
5.23.3 Local CNG Synthesis 120
5.23.4 Memory Update in CNG-ER and CNG-QR 121
6 Functional description of the decoder 122
6.1 Reconstruction of the Excitation 122
6.2 Excitation Post-processing 124
6.2.1 Anti-Sparseness Processing in Generic HR 124
6.2.2 Gain Smoothing for Noise Enhancement 125
6.2.3 Pitch Enhancer for Generic and Voiced Encoding Types 126
6.3 Synthesis, Post-processing and Up-sampling 126
6.3.1 Low-Frequency Pitch Enhancement Post-processing 127
6.3.2 High-Pass Filtering 129
6.4 Reconstruction of High-Frequency Band 130
6.4.1 Generation of High-Band Excitation 130
6.4.2 LP Filter for the High-Frequency Band 131
6.4.3 High-Band Synthesis 131
6.5 Frame Error Concealment 132
6.5.1 Construction of the Periodic Part of the Excitation 133
6.5.2 Construction of the Random Part of the Excitation 134
6.5.3 Spectral Envelope Concealment, Synthesis and Updates 134
6.5.4 Recovery of Normal Operation after an Erasure 135
6.5.4.1 Artificial Onset Reconstruction 135
6.5.5 Energy Control 136
6.6 Decoding of Inactive Speech Frames (CNG-ER and CNG-QR) 138
6.7 Detection and Concealment of Frames with Corrupted Rate Information 139
6.7.1 Test of Frame Structure 139
6.7.2 CNG Frames 139
6.7.2.1 Test of the ISF Ordering 140
6.7.2.2 Test of the Variation of the Quantized Energy 140
6.7.3 Active Speech Frames 140
6.7.3.1 Test of the ISF Ordering 141
6.7.3.2 Test of the LP Gain against the Fixed-Codebook Gain 141
6.7.3.3 Test of the Synthesis Energy against the FER Energy Information 142
7 Interoperable interconnection between VMR-WB and AMR-WB codecs 144
7.1 VMR-WB to AMR-WB Interconnection (Reverse Link) 144
7.2 AMR-WB to VMR-WB Interconnection (Forward Link) 146
8 VMR-WB frame structure 148

8.1 Frame Structure of Full-Rate Encoding Types ... 148
8.1.1 Frame Structure of Generic Full-Rate ... 148
8.1.2 Frame Structure of 12.65 kbps Interoperable Full-Rate ... 150
8.1.3 Frame Structure of 8.85 kbps Interoperable Full-Rate ... 151
8.1.4 Frame Structure of 6.60 kbps Interoperable Full-Rate ... 152
8.2 Frame Structure of Half-Rate Encoding Types ... 154
8.2.1 Frame Structure of Generic Half-Rate ... 154
8.2.2 Frame Structure of Signaling Half-Rate ... 155
8.2.3 Frame Structure of Voiced Half-Rate ... 155
8.2.4 Frame Structure of Unvoiced Half-Rate ... 156
8.2.5 Frame Structure of 12.65 kbps Interoperable Half-Rate ... 157
8.2.6 Frame Structure of 8.85 kbps Interoperable Half-Rate ... 157
8.2.7 Frame Structure of 6.60 kbps Interoperable Half-Rate ... 158
8.3 Frame Structure of Quarter-Rate Encoding Types ... 158
8.3.1 Frame Structure of CNG Quarter-Rate ... 158
8.3.2 Frame Structure of Unvoiced Quarter-Rate ... 159
8.4 Frame Structure of CNG Eighth-Rate ... 159
8.5 MIME/File Storage Format ... 160
8.5.1 Single channel Header ... 160
8.5.2 Multi-channel Header (Currently not Implemented) ... 160
8.5.3 Speech Frames ... 161
9 Support for TDD/TTY and low-rate In-band data ... 163
9.1 TTY/TDD Frame Format ... 163
9.1.1 Rate 1 with TDD/TTY Data ... 163
9.1.2 Full-Rate with TDD/TTY Data + Speech Data ... 163
9.1.3 Half-Rate with TDD/TTY Data ... 164

1 SOURCE-CONTROLLED VARIABLE-RATE MULTIMODE TWO-WAY WIDEBAND VOICE COMMUNICATION (VMR-WB)

1.1 General Description

Service Option 62, the source-controlled variable-rate multimode wideband speech codec (VMR-WB), provides two-way voice communication between the base station and the mobile station using the dynamically variable data rate speech codec algorithm described in this standard. The transmitting speech codec receives voice samples and generates an encoded speech packet for every Traffic Channel frame*. The receiving station generates a speech packet from every Traffic Channel frame and supplies it to the speech codec for decoding into voice samples.

It should be noted that the contents of this document describe all operational modes of the VMR-WB codec inclusive of the AMR-WB interoperable mode and its associated interworking functions. While both Service Option 62 and the AMR-WB interoperable mode share the same algorithmic description, Service Option 62 provides methods for initialization and call setup using VMR-WB modes 0, 1, 2, and 2 with maximum half-rate. Furthermore Service Option 62 shall be the primary Service Option for circuit-switched wideband voice calls in cdma2000® terminals that support VMR-WB codec. The applications and call setup using the AMR-WB interoperable mode are beyond the scope of this document.

VMR-WB communicates at one of four rates: 13300 bps, 6200 bps, 2700 bps and 1000 bps corresponding to Rate-Set II of CDMA systems.

During an interoperable interconnection using VMR-WB mode 3, the operation bandwidth shall be wideband and switching to other modes shall not be allowed.

All implementations shall meet the minimum performance requirements defined in 3GPP2 C.S0053-0.

1.2 Overview of VMR-WB Documentation

The VMR-WB specification family consists of two standards. This standard provides the algorithmic description of the VMR-WB as well as the master floating-point C-simulation of the codec. The companion minimum performance standard, “Recommended Minimum Performance Standard for the Variable-Rate Multimode Wideband Speech Codec, Service Option 62 ”, consists of the master fixed-point C-simulation of the VMR-WB codec as well as minimum performance specification and the associated test vectors and processing tools. The minimum performance specification further consists of a set of objective and subjective tests used to verify the quality of any non bit-exact VMR-WB implementation.

---

* 3GPP2 C.S0003-0 uses the term “frame” to represent a 20 ms grouping of data on the Fundamental Channel. Common speech codec terminology also uses the term “frame” to represent a quantum of processing. For Service Option 62, the speech codec frame corresponds to speech sampled over 20 ms. The speech samples are processed into a packet. This packet is transmitted in a Traffic Channel frame.

cdma2000® is the trademark for the technical nomenclature for certain specifications and standards of the Organizational Partners (OPs) of 3GPP2. Geographically (and as of the date of publication), cdma2000® is a registered trademark of the Telecommunications Industry Association (TIA–USA) in the United States.

Section 2 of this standard provides information on the interfaces between VMR-WB and generic cdma2000 air-interface (i.e., these technical specifications do not rely upon a particular version of cdma2000 air-interface). Section 3 provides information on the audio interfaces to VMR-WB. The specifications given by Sections 4, 5, and 6 of this standard provide the algorithmic description of VMR-WB. The necessary interworking functions for establishing an interoperable interconnection between VMR-WB and AMR-WB are described in Section 7. Section 8 provides the detailed description of VMR-WB frame structure and MIME/file storage format. The support for TDD/TTY and low rate in-band data is described in Section 9.

The floating-point C-code accompanying this document provides a more detailed complementary description of VMR-WB. In the case of a discrepancy between the floating-point C-simulation and the algorithmic description, the floating-point C-simulation will prevail.

1.3 VMR-WB Implementation Options

There are two options available for implementation of the VMR-WB:

If a bit-exact implementation approach is chosen, the implementation that is developed shall be end-to-end bit-exact to the reference fixed-point C-simulation. Bit-exactness is verified via application of the associated VMR-WB minimum performance standard 3GPP2 C.S0053-0, which includes the reference fixed-point codec simulation as well as a number of test vectors.

Alternatively, an implementation that deviates from the end-to-end bit-exact VMR-WB specification described above may be developed. Such an implementation shall pass the objective and subjective tests defined by the VMR-WB minimum performance standard 3GPP2 C.S0053-0.

1.4 VMR-WB Algorithmic Options

By default, VMR-WB is interoperable with 3GPP/AMR-WB (ITU-T/G.722.2) only at 12.65 kbps in mode 3. However, the interoperability can be expanded to 3GPP/AMR-WB (ITU-T/G.722.2) at 6.60 and 8.85 kbps by selecting optional compilation flag, EXPANDED_INTEROPERABILITY, in the VMR-WB C simulation.

While in the AMR-WB interoperable mode, mode switching is not allowed. There is only one AMR-WB interoperable mode in VMR-WB. During an AMR-WB interoperable interconnection, the AMR-WB codec on the GSM/WCDMA side depending on channel conditions may request VMR-WB encoder to switch between AMR-WB codec modes 0, 1, and 2 corresponding to 6.60, 8.85, and 12.65 kbps. In-band data embedded in VMR-WB frame structure, as shown in Section 8, is used during an AMR-WB interoperable interconnection to switch between AMR-WB codec modes 0, 1, or 2 without changing the operation mode of VMR-WB. Other AMR-WB codec modes that are not specified in this document are not supported in VMR-WB.

The encoding and decoding procedures of the optional AMR-WB 8.85 and 6.60 kbps codec modes in VMR-WB C simulation are identical with the corresponding modes in AMR-WB codec. The algorithmic description of the optional AMR-WB 8.85 and 6.60 kbps codec modes are not included in this document and can be found in 3GPP/AMR-WB specifications [9,13,14,16].

By default, the VMR-WB codec is capable of detection and concealment of frames with corrupted rate information. However, the sensitivity of the bad rate detection algorithm can be increased by selecting the compilation option BRH_LEVEL2 in the C simulation of the VMR-WB codec (see Section 6.7).

1.5 Service Option Number

The variable-rate multimode two-way wideband/narrowband voice Service Option, using the speech codec algorithm described by this standard, shall use Service Option number 62, for initialization and call set up using VMR-WB modes 0, 1, 2, and 2 with maximum half-rate, and shall be called Service Option 62.

1.6 Allowable Delays

1.6.1 Allowable Transmitting Speech Codec Encoding Delay

The transmitting speech codec shall supply a packet to the multiplex sublayer no later than 20 ms after it has obtained the last input sample for the current speech frame.

1.6.2 Allowable Receiving Speech Codec Decoding Delay

The receiving decoder shall generate the first sample of speech using parameters from a packet supplied to it by the multiplex sublayer no later than 3 ms after being supplied the packet.

1.7 Special Cases

1.7.1 Blanked Packets

A blanked frame occurs when the transmitting station uses the entire frame for either signaling traffic or secondary traffic. The VMR-WB encoder does no special encoding process during the generation of a blank packet; i.e., the generated voice packet is simply not used. The decoder, in turn, treats a blank packet in the same manner as a frame erasure.

1.7.2 Null Traffic Channel Data

A Rate 1/8 packet with all bits set to ‘1’ is considered as null Traffic Channel data. This packet is declared an erased packet and handled as described in Section 6.5. If more than two consecutive all-ones Rate 1/8 packets are received, the decoder’s output shall be muted until a valid packet is received.

1.7.3 All Zeros Packet

Rate 1, Rate ½, Rate ¼, and Rate 1/8 packets with all bits set to ‘0’ shall be considered erased frames by the decoder and shall be handled as described in Section 6.5.

1.8 Terms and Numeric Information

Adaptive Codebook (ACB). The adaptive codebook contains excitation vectors that are adapted for every subframe. The adaptive codebook is derived from the long-term filter state. The lag value can be viewed as an index into the adaptive codebook.

Algebraic Codebook. A fixed-codebook where algebraic code is used to populate the excitation vectors (innovation vectors). The excitation contains a small number of nonzero pulses with predefined interlaced sets of potential positions. The amplitudes and positions of the pulses of the *k*th excitation codevector can be derived from its index *k* through a rule requiring no or minimal physical storage, in contrast with stochastic codebooks, whereby the path from the index to the associated codevector involves look-up tables.

Algebraic Code Excited Linear Predictive Coding (ACELP). The algorithm that is used by the encoder to generate the stochastic component of the excitation where an excitation vector contains a few number of non-zero pulses with predefined interlaced sets of positions. The pulses have their amplitudes fixed to +1 or -1, and each pulse has a set of possible positions distinct from the positions of the other pulses. The set of positions are interlaced. The excitation code is identified by the positions of its non-zero pulses. The codebook search is essentially searching the optimum position of the non-zero pulses.

Anti-Sparseness Processing. An adaptive post-processing procedure applied to the fixed-codebook vector in order to reduce perceptual artifacts from a sparse fixed-codebook vector.

Autocorrelation Function. A function showing the relationship of a signal with a time-shifted version of itself.

Base Station. A station in the Public Radio Telecommunications Service, other than a personal/mobile station, used for radio communications with personal/mobile stations.

Closed-Loop Pitch Analysis: This is the adaptive codebook search, i.e. a process of estimating the pitch (lag) value from the weighted input speech and the long-term filter state. In the closed-loop search, the lag is searched using error minimization loop (analysis-by-synthesis). In the VMR-WB codec, closed-loop pitch search is performed for every subframe.

Codec. The combination of an encoder and decoder in series (encoder/decoder).

Code Excited Linear Predictive Coding (CELP). A speech-coding algorithm, CELP coders use codebook excitation, a long-term pitch prediction filter, and a short-term formant prediction filter.

Codebook. A set of vectors used by the speech codec. For each speech codec codebook subframe, one particular vector is chosen and is used to excite the speech codec's filters. The codebook vector is chosen to minimize the weighted error between the original and synthesized speech after the pitch and formant synthesis filter coefficients have been determined.

Coder. Same as "encoder."

Decoder. Generally, a device for the translation of a signal from a digital representation into an analog format. For this standard, a device that converts speech encoded in the format specified in this standard to analog or an equivalent PCM representation.

Encoder. Generally, a device for the translation of a signal into a digital representation. For this standard, a device that converts speech from an analog, or from its equivalent PCM representation, to the digital representation described in this standard.

Formant. A resonant frequency of the human vocal tract causing a peak in the short-term spectrum of speech.

Fractional Lags. A set of lag values having sub-sample resolution. In VMR-WB codec a sub-sample resolution of 1/4th or 1/2nd of a sample is used.

IIR Filter. (Infinite-duration impulse response filter) A filter for which the output, in response to an impulse input, never totally converges to zero. This term is usually used in reference to digital filters.

Interpolating Filter. An FIR filter used to produce an estimate of sub-sample resolution samples, given an input sampled with integer sample resolution. In this implementation, the interpolating filter has low-pass filter characteristics. Thus the adaptive codebook consists of the low-pass filtered interpolated past excitation.

Inverse Filter: This filter removes the short-term correlation from the speech signal. The filter models an inverse frequency response of the vocal tract.

Lag: The long term filter delay. This is typically the true pitch period, or its multiple or submultiples.

LP Analysis Window: For each frame, the short-term filter coefficients are computed using the high pass filtered speech samples within the analysis window. In VMR-WB codec, the length of the analysis window is always 384 samples. For all the modes, a single asymmetric window is used to generate a single set of LP coefficients. The 10 ms look-ahead is used in the analysis.

Linear Predictive Coding (LPC). A method of predicting future samples of a sequence by a linear combination of the previous samples of the same sequence. Linear Predictive Coding is frequently used in reference to a class of speech codecs.

Immitance Spectral Frequencies (ISFs). A representation of digital filter coefficients in a pseudo-frequency domain. This representation has good quantization and interpolation properties.

LSB. Least significant bit.

Mode. An operating condition of the codec that corresponds to certain average data rate and subjective quality.

MSB. Most significant bit.

Normalized Autocorrelation Function. A measure used to determine the pitch period and the degree of periodicity of the input speech. This measure is useful in distinguishing voiced from unvoiced speech.

Open-Loop Pitch Search: A process of estimating the near optimal lag directly from the weighted speech input. This is done to simplify the pitch analysis and confine the closed-loop pitch search to a small number of lags around the open-loop estimated lags.

Packet. The unit of information exchanged between Service Option applications in the base station and the personal/mobile station.

Perceptual Weighting Filter: This filter is employed in the analysis-by-synthesis search of the codebooks. The filter exploits the noise masking properties of the formants (vocal tract resonances) by weighting the error less in regions near the formant frequencies and more in regions away from them.

Personal/Mobile Station. A station in the Public Radio Telecommunications Service intended to be used while in motion or during halts at unspecified points.

Pitch. The fundamental frequency in speech caused by the periodic vibration of the human vocal cords.

RDA. Rate Determination Algorithm.

Relaxation Code Excited Linear Predictive Coding (RCELP). The speech coding algorithm used by the encoder where unlike conventional CELP coders, a modified version of the speech signal that conforms to a linearly interpolated pitch contour is encoded, relaxing the frequent pitch update constraint in low rate CELP coders. This pitch contour is obtained by estimating the pitch values at the analysis frame boundaries and linearly interpolating the pitch across frame.

Residual. The output signal resulting from an inverse filtering operation

RLR. Receive Loudness Rating, a measure of receive audio sensitivity, as defined in IEEE Standard 269-2002. The measurement of the receive loudness rating is described in ITU-T Recommendation P.79-1999.

Short-Term Synthesis Filter: This filter introduces short-term correlation into the excitation signal, which models the impulse response of the vocal tract.

SLR. Send Loudness Rating, a measure of transmit audio sensitivity, as defined in IEEE Standard 269-2002. The measurement of the send loudness rating is described in ITU-T Recommendation P.79-1999.

SPL. Sound Pressure Level.

Subframe: A time interval equal to 5 ms (80 samples at 16 kHz sampling rate).

Vector Quantization: A method of grouping several parameters into a vector and quantizing them simultaneously.

Voiced Speech. Speech generated when the vocal cords are vibrating at a fundamental frequency. Characterized by high energy, periodicity, and a large ratio of energy below 2 kHz to energy above 2 kHz.

Unvoiced Speech. Speech generated by forcing air through constrictions in the vocal tract without vibration of the vocal cords. Characterized by a lack of periodicity, and a near-unity ratio of energy below 2 kHz to energy above 2 kHz.

WAEPL. Weighted Acoustic Echo Path Loss. A measure of the echo performance under normal conversation. ANSI/EIA/TIA-579-A98 defines the measurement of WAEPL.

Zero Input Response (ZIR). The filter output caused by the non-zero initial state of the filter when no input is present.

Zero State Response (ZSR). The filter output caused by an input when the initial state of the filter is zero.

ZIR. See Zero Input Response.

ZSR. See Zero State Response.

2 REQUIRED MULTIPLEX OPTION SUPPORT

Service Option 62 shall support an interface with Multiplex Option 2. Speech packets for Service Option 62 shall only be transported as primary traffic. Service Option 62 shall be the primary Service Option for circuit-switched wideband voice calls in cdma2000 terminals that support VMR-WB codec.

2.1 Interface to Multiplex Option 2

2.1.1 Transmitted Packets

The speech codec shall generate and shall supply exactly one packet to the multiplex sublayer every 20 milliseconds. The packet contains the Service Option information bits that are transmitted as primary traffic. The packet shall be one of five types as shown in Table 2.1-1. The number of bits supplied to the multiplex sublayer for each type of packet shall also be as shown in Table 2.1-1. Unless otherwise commanded, the speech codec may supply a Rate 1, Rate 1/2, Rate 1/4, or Rate 1/8 packet. Upon command, the speech codec shall generate a Blank packet. Also upon command, the speech codec shall generate a non-blank packet with a maximum rate of Rate 1/2.

A Blank packet contains no bits and is used for blank-and-burst transmission of signaling traffic or secondary traffic (see 3GPP2 C.S0003-0 and 3GPP2 C.S0005-0†).

Table 2.1-1. Packet Types Supplied by Service Option 62 to the Multiplex Sublayer

Packet Type	Bits per Packet
Rate 1	266
Rate 1/2	124
Rate 1/4	54
Rate 1/8	20
Blank	0

2.1.2 Received Packets

The multiplex sublayer in the receiving station categorizes every received Traffic Channel frame, and supplies the packet type and accompanying bits, if any, to the speech codec as shown in Table 2.1-1. The speech codec processes the bits of the packet as described in Sections 6 and 7. The received packet types shown in Table 2.1-2 correspond to the transmitted packet types shown in Table 2.1-1. The Blank packet type occurs when the receiving station determines that a blank-and-burst frame for signaling traffic or secondary traffic was transmitted. When the multiplex sublayer determines that a received frame is in error, the multiplex sublayer supplies an insufficient frame quality (erasure) packet to the Service Option 62.

† The technical specifications described in this document do not rely upon a particular version of cdma2000® air-interface.

Table 2.1-2. Packet Types Supplied by the Multiplex Sublayer to Service Option 62

Packet Type	Bits per Packet
Rate 1	266
Rate 1/2	124
Rate 1/4	54
Rate 1/8	20
Blank	0
Insufficient frame quality (erasure)	0

2.2 Negotiation for Service Option 62

The mobile station and base station can negotiate for Service Option 62 service negotiation, as described in 3GPP2 C.S0005-0 [7]. This section describes the service negotiation and call set up for modes 0, 1, 2, and 2 with maximum half-rate of VMR-WB standard.

2.2.1 Procedures Using Service Negotiation

The mobile station and base station shall perform service negotiation for Service Option 62 as described in 3GPP2 C.S0005-0 [7], and the negotiated service configuration shall include only valid attributes for the Service Option as specified in Table 2.2-1.

Table 2.2-1. Valid Service Configuration Attributes for Service Option 62

Service Configuration Attribute	Valid Selections
Forward Multiplex Option	Multiplex Option 2
Reverse Multiplex Option	Multiplex Option 2
Forward Transmission Rates	Rate Set 2 with all rates enabled
Reverse Transmission Rates	Rate Set 2 with all rates enabled
Forward Traffic Type	Primary Traffic
Reverse Traffic Type	Primary Traffic

2.2.1.1 Initialization and Connection

2.2.1.1.1 Mobile Station Requirements

If the mobile station accepts a service configuration, as specified in a *Service Connect Message*, *General Handoff Direction Message*, or *Universal Handoff Direction Message* that includes a Service Option connection using Service Option 62, the mobile station shall perform the following:

- If the Service Option connection is new (that is, not part of the previous service configuration), the mobile station shall perform speech codec initialization at the time specified by the maximum of the action time associated with the message carrying the Service Configuration Record, and the time that the corresponding Call Control Instance is instantiated. The mobile station shall initialize its VMR-WB encoder mode of operation to a default value of 0 and the operational bandwidth to wideband. The mobile station shall complete the initialization within 40 ms.
- Beginning at the time specified by the maximum of the action time associated with the message carrying the Service Configuration Record, and the time that the corresponding Call Control Instance is instantiated, and continuing for as long as the service configuration includes the

Service Option connection, Service Option 62 shall process received packets and generate and supply packets for transmission as follows:

- If the Call Control Instance is in the *Conversation Substate*, Service Option 62 shall process the received packets and generate and supply packets for transmission in accordance with this standard.
- If the Call Control Instance is not in the *Conversation Substate*, Service Option 62 shall process the received packets in accordance with this standard, and shall generate and supply Rate 1/8 Packets with all bits set to '1' for transmission, except when commanded to generate a Blank packet.

2.2.1.1.2 Base Station Requirements

If the base station establishes a service configuration, as specified in a *Service Connect Message*, *General Handoff Direction Message*, or *Universal Handoff Direction Message* that includes a Service Option connection using Service Option 62, the base station shall perform the following:

- If the Service Option connection is new (that is, not part of the previous service configuration), the base station shall perform speech codec initialization no later than the time specified by the maximum of the action time associated with the message carrying the Service Configuration Record, and the time that the corresponding Call Control Instance is Instantiated. The base station shall initialize its VMR-WB encoder mode of operation to a default value of 0 and the operational bandwidth to wideband.
- Commencing at the time specified by the maximum of the action time associated with the message carrying the Service Configuration Record, and the time that the corresponding Call Control Instance is Instantiated, and continuing for as long as the service configuration includes the Service Option connection, Service Option 62 shall process received packets, and shall generate and supply packets for transmission in accordance with this standard. The base station may defer enabling the audio input and output.

2.2.1.2 Service Option Control Messages

2.2.1.2.1 Mobile Station Requirements

The mobile station shall support one pending *Service Option Control Message* for Service Option 62.

If the mobile station receives a *Service Option Control Message* for Service Option 62, then, at the action time associated with the message, the mobile station shall process the message as follows:

1. If the MOBILE_TO_MOBILE field is equal to '1', the mobile station should disable the audio output of the speech codec for 1 second after initialization.

   If the MOBILE_TO_MOBILE field is equal to '0', the mobile station shall process each received packet as described in Section 6.
2. If the INIT_CODEC field is equal to '1', the mobile station shall perform speech codec initialization. The mobile station shall complete the initialization within 40 ms.
3. VMR-WB accepts as input, a mode of operation through the RATE_REDUC field as defined in Table 2.2-2 using this mode of operation, VMR-WB generates Rate 1, Rate ½, Rate ¼, and Rate 1/8 packets in a proportion that results in the average data rate given by Table 2.2-2.

Service Option 62 shall continue to use these fractions until either of the following events occurs:

- The mobile station receives a *Service Option Control Message* specifying a different RATE_REDUC, or
- Service Option 62 is initialized.

Service Option 62 was developed using reduced rate operation (encoding mode of operation) as a network control criteria. The VMR-WB codec selects the encoding rate based upon the instantaneous characteristics of the input speech: voiced, unvoiced, transition, etc., as well as the encoding mode selected.

While dynamic mode switching is allowed between all modes associated with Service Option 62 with a minimum mode-switching period of 20ms, change of operational bandwidth is not recommended during a conversation. The default operation bandwidth is wideband. Once the operational bandwidth is specified, the mode of operation can be dynamically switched during a conversation.

Table 2.2-2. VMR-WB Encoding Rate Mode Control Parameters

RATE_REDUC (Binary)	Operational Bandwidth	VMR-WB Mode of Operation	Estimated Average Encoding Rate kbps (Source Encoding Rates)
‘000’	Wideband	0	9.1404
‘001’		1	7.6930
‘010’		2	6.2847
‘011’		2 (HR Max)	4.7443
‘100’	Narrowband	0	9.0435
‘101’		1	7.5276
‘110’		2	6.2109
‘111’		2 (HR Max)	4.7518

2.2.1.2.2 Base Station Requirements

The base station may send a *Service Option Control Message* to the mobile station. If the base station sends a *Service Option Control Message*, the base station shall include the following type-specific fields for Service Option 62:

Table 2.2-3. Service Option Control Message Type-Specific Fields

Field	Length (bits)
RATE_REDUC	3
RESERVED	3
MOBILE_TO_MOBILE	1
INIT_CODEC	1

RATE_REDUC - VMR-WB mode of operation.

The base station shall set this field to the RATE_REDUC value from Table 2.2-2 corresponding to the mode of operation that the mobile station is to operate in.

RESERVED - Reserved bits.

The base station shall set this field to ‘000’.

MOBILE_TO_MOBILE

- Mobile-to-mobile processing.

If the mobile station is to perform mobile-to-mobile processing (Section 2.2.1.2.1), the base station shall set this field to '1'. In addition, if the mobile station is to disable the audio output of the speech codec for 1 second after initialization, the base station shall set the INIT_CODEC field and the MOBILE_TO_MOBILE field to '1'. If the mobile station is not to perform mobile-to-mobile processing, the base station shall set the MOBILE_TO_MOBILE field to '0'.

INIT_CODEC - Initialize speech codec.
If the mobile station is to initialize the speech codec, the base station shall set this field to '1'. Otherwise, the base station shall set this field to '0'.

Table 2.2-4. VMR-WB Encoding Rate Mode Control Parameters

RATE_REDUC (Binary)	Operational Bandwidth	VMR-WB Mode of Operation	Estimated Average Encoding Rate kbps (Source Encoding Rates)
'000'	Wideband	0	9.1404
'001'		1	7.6930
'010'		2	6.2847
'011'		2 (HR Max)	4.7443
'100'	Narrowband	0	9.0435
'101'		1	7.5276
'110'		2	6.2109
'111'		2 (HR Max)	4.7518

3 AUDIO INTERFACES

In general the basic audio interface characteristics of VMR-WB codec are shown in Table 3-1.

Table 3-1: Basic audio interface characteristics of VMR-WB

Bandwidth	50 – 7000 Hz (-3dB) @ 16 kHz sampling frequency for wideband operation 100 – 3700 Hz (-3dB) @ 8 kHz sampling frequency for narrowband operation
Nominal Input Speech Level	-25 dBov per ITU-T P.56 Speech Voltmeter
Recommended Maximum Input Speech Level Variation	-15 to -35 dBov per ITU-T P.56 Speech Voltmeter
Minimum Resolution	13 bit linear, 2's complement PCM
Recommended SNR	Greater than 10 dB

3.1 Input Audio Interface

The analog-to-digital and digital-to-analog conversion will in principle comprise the following elements:

1) Analog to uniform digital PCM
- Microphone;
- Input level adjustment device;
- Input anti-aliasing filter;
- Sample-hold device sampling at 16 kHz;
- Analog-to-uniform digital conversion to 16-bit representation.

The uniform format shall be represented in two's complement.

2) Uniform digital PCM to analog
- Conversion from 16-bit/16 kHz uniform PCM to analog;
- A hold device;
- Reconstruction filter including x/sin (x) correction;
- Output level adjustment device;
- Earphone or loudspeaker.

In the terminal equipment, the A/D function may be achieved by direct conversion to 16-bit uniform PCM format; For the D/A operation, the inverse operations take place.

Lower uniform PCM precisions such as 14-bit should be adjusted to 16-bit representation by magnitude scaling; i.e., shifting the bits to the left and setting the LSBs to zero.

3.1.1 Input Audio Interface in the Mobile Station

The input audio may be either an analog or digital signal.

3.1.1.1 Conversion and Scaling

Whether the input is analog or digital by default, the signal presented to the input of the wideband speech codec should span a frequency range of 50-7000 Hz and shall be sampled at a rate of 16000 samples per second and should be quantized to a uniform PCM format with 16 bits of dynamic range.

The wideband codec is also capable of processing narrowband speech signals (100-3700 Hz) sampled at 8000 Hz with recommended precision of 16-bits and reproduce narrowband outputs sampled at 8000 Hz with the same precision.

The default input sampling rate for the encoder and the decoder of VMR-WB is 16000 Hz. As appropriate, an input/output sampling rate of 8000 Hz may be signaled to the encoder and the decoder when they are initialized.

The quantities in this standard assume 16-bit integer input normalization with a range from -32,768 through +32,767. The following speech codec discussion assumes this 16-bit integer normalization. If an input audio interface uses a different normalization scheme, then appropriate scaling should be used.

3.1.1.2 Digital Audio Input

If the input audio is an 8-bit μ-Law/A-Law PCM signal, it should be converted to a uniform PCM format according to Table 2 for μ-Law and Table 1 for A-Law in ITU-T Recommendation G.711 “Pulse Code Modulation (PCM) of Voice Frequencies. After this conversion, the uniform PCM signal should be scaled in magnitude by shifting to the left by 2 bits (3 bits for A-Law) and setting the 2 (3 for A-Law) LSBs to zero to form a 16-bit integer. This will ensure normalization of the 8-bit input signal to the overload point of the 16-bit linear quantization.

3.1.1.3 Analog Audio Input

If the input is in analog form, the mobile station should sample the analog speech and should convert the samples to a digital format for speech codec processing. This may be accomplished by either the following or an equivalent method. First, the input gain audio level is adjusted. Then, the signal is bandpass filtered to prevent aliasing. Finally, the filtered signal is sampled and quantized (Section 3.1.1.1).

3.1.1.3.1 Transmit Level Adjustment

The mobile station should have a Send Loudness Rating (SLR) equal to 11 ±3dB, when transmitting to a reference base station. The send loudness ratings are described in ITU-T Recommendation P.79-1999 “Calculation of loudness ratings for telephone sets”.

3.1.1.3.2 Band Pass Filtering

If the input speech is wideband, the signal should be filtered by a filter that generally conforms to the mask shown in ITU-T G.722 (11/1998) Figure 10/G.722, “Attenuation Distortion vs. Frequency”.

If the input speech is narrowband, the signal should be filtered by a filter that generally conforms to the mask shown in ITU-T G.712 (11/2001) Figure 4/G.712, “Attenuation/frequency distortion for channels between a 4-wire analog port and a digital port ($E4_{in}$ to $T_{ou}t$)”.

The manufacturer may provide additional anti-aliasing filtering.

3.1.1.3.3 Echo Return Loss

Provision shall be made to ensure adequate isolation between receive and transmit audio paths in all modes of operation. When no external transmit audio is present, the speech codec should not generate packets at rates higher than Rate 1/8 due to acoustic coupling of the receive audio into the transmit audio path (specifically with the receive audio at full volume). Minimum target levels of 45 dB

WAEPL should be met. See ANSI/EIA/TIA-579-A-98, *Telecommunications Telephone Terminal Equipment Transmission Requirements for Digital Wireline Telephones*, Nov. 1998.

3.1.2 Input Audio Interface in the Base Station

3.1.2.1 Sampling and Format Conversion

The base station converts the input speech (analog, μ-Law/A-Law companded Pulse Code Modulation, or other format) into a uniform quantized PCM format with 16 bits of dynamic range. The sampling rate by default is 16000 samples per second, unless narrowband processing is desired. In the case of narrowband input/output speech processing, the sampling rate is 8000 samples per second. The sampling and conversion process should be as in Section 3.1.1.1.

3.1.2.2 Transmit Level Adjust

The base station should set the transmit level so that a 1004 Hz tone at a level of 0 dBm0 at the network interface produces a level 3.17 dB below maximum amplitude at the output of the quantizer.

3.1.2.3 Line Echo Canceling

In case of narrowband operation, the base station should provide a method to cancel echoes returned by the PSTN interface[†]. The echo canceling function should meet ITU-T G.168 requirement.

3.2 Output Audio Interface

3.2.1 Output Audio Interface in the Mobile Station

3.2.1.1 Band Pass Filtering

If the output speech is wideband, the signal should be filtered by a filter that generally conforms to the mask shown in ITU-T G.722 (11/1998) Figure 10/G.722, “Attenuation Distortion vs. Frequency”.

If the output speech is narrowband, the signal should be filtered by a filter that generally conforms to the mask shown in ITU-T G.712 (11/2001) Figure 4/G.712, “Attenuation/frequency distortion for channels between a 4-wire analog port and a digital port ($E4_{in}$ to $T_{ou}t$)”.

The manufacturer may provide additional reconstruction filtering.

3.2.1.2 Receive Level Adjustment

The mobile station should have a nominal Receive Loudness Rating (RLR) equal to 3 ±3 dB when receiving from a reference base station. The receive loudness ratings are described in ITU-T Recommendation P.79-1999 “Calculation of loudness ratings for telephone sets”.

3.2.2 Output Audio Interface in the Base Station

Details of the digital and analog interfaces to the network are outside the scope of this document.

3.2.2.1 Receive Level Adjustment

The base station should set the audio level so that a received 1004 Hz tone 3.17 dB below maximum amplitude produces a level of 0 dBm0 at the network interface.

[†] Because of the relatively long delays inherent in the speech coding and transmitting processes, echoes that are not sufficiently suppressed are noticeable to the mobile station user.

4 THE VARIABLE-RATE MULTIMODE WIDEBAND SPEECH CODEC (VMR-WB) – INTRODUCTION AND BIT ALLOCATION TABLES

4.1 Introduction to the VMR-WB Speech Coding Algorithm

VMR-WB is a source-controlled variable-rate multimode codec designed for encoding/decoding of wideband speech (50-7000 Hz). It is based on 3GPP/AMR-WB (ITU-T/G.722.2) core technology [9]. VMR-WB is fully interoperable with both standards at 12.65, 8.85*, and 6.60* kbps in the AMR-WB interoperable mode of operation.

The VMR-WB algorithm is based on that of AMR-WB at 12.65 kbps; however, it is optimized to operate efficiently in the cdma2000 system by using a source-controlled variable-bit-rate paradigm and the addition of Half-Rate (HR), Quarter-Rate (QR), and Eighth-Rate (ER) encoding schemes to achieve the best subjective quality at various average data rates (ADR).

The operation of VMR-WB is controlled by speech signal characteristics (i.e., source-controlled) and by traffic condition of the network (i.e., network-controlled mode switching). Depending on the traffic conditions, one of 4 operational modes is used. Modes 0, 1, and 2 are specific to CDMA systems (i.e., cdmaOne, cdma2000) with mode 0 providing the highest quality and mode 2 the lowest ADR. Mode 3 is the AMR-WB interoperable mode operating at an ADR slightly higher than Mode 0 and providing a quality equal or better than that of AMR-WB at 12.65 kbps when in an interoperable interconnection with AMR-WB at 12.65 kbps.

There are a number of various encoding types used in the VMR-WB codec. The 12.65/8.85/6.60 kbps Interoperable Full-Rate (FR) types are interoperable with AMR-WB at 12.65, 8.85, and 6.60 kbps, respectively. In the Generic FR type at 13.3 kbps, the extra 13 bits (i.e., the difference between 12.65 and 13.3 kbps) are used to enhance the codec performance in frame erasure conditions. Thus, the codec can interoperate with ITU-T G.722.2/AMR-WB at 12.65, 8.85, and 6.60 kbps without compromising the quality and performance in the cdma2000 network.

The VMR-WB coding system consists of a variable-rate multimode encoder, a decoder, and three interworking functions, each operating at bit-stream level (i.e. no decoding is necessary). The first interworking function, associated with Service Option 62, allows the use of Dim-and-Burst signaling during VMR-WB mobile-to-mobile calls (e.g., in TFO), or in other cases where a signaling request for generation of a Half Rate frame type may not be communicated to the originating VMR-WB encoder. In this case, Full-Rate frames that arrive at the multiplex sub-layer coincident with signaling requests shall be reduced from Full-Rate to Half-Rate to accommodate the signaling insertion. Frames that arrive at the multiplex sub-layer during non-signaling frames, or frames that are sufficiently low in rate to allow Dim-and-Burst signaling shall not be modified.

The frame rate reduction process (or "packet-level signaling") is facilitated by the Half-Rate Signaling encoding type. This is substantially a Full-Rate frame with certain parameter bit fields stripped away. This is shown in more detail later in Table 4.2-1 and Table 4.2-2. Figure 4.1-1 shows a block diagram of the use of this interworking function.

---

* By default VMR-WB is only interoperable with AMR-WB at 12.65 kbps; however, activating compilation flag EXPANDED_INTEROPERABILITY in the C simulation would allow interoperability with AMR-WB at 6.60 and 8.85 kbps, as well.

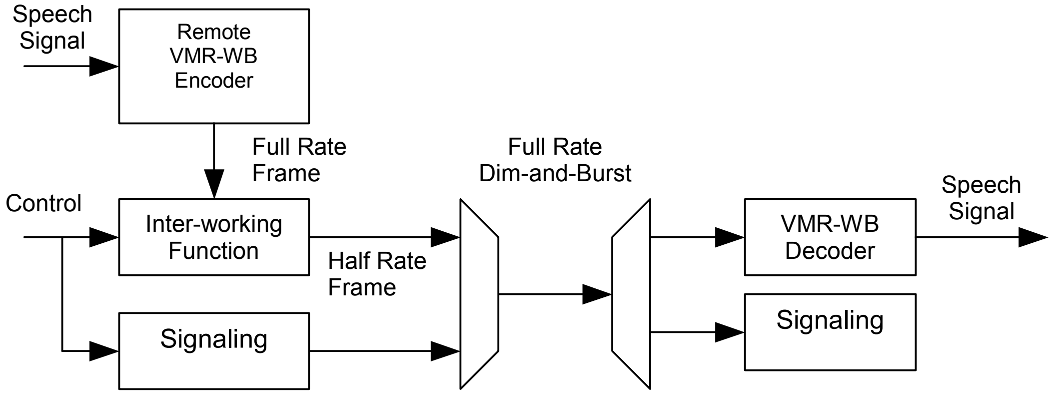

Figure 4.1-1: Functional block-diagram of the first Interworking Function

The second interworking function allows for bit-stream conversion from AMR-WB at 12.65 kbps (also 8.85 or 6.60 kbps) to VMR-WB and for inserting CDMA signaling information if necessary (as above). The third interworking function enables bit-stream conversion from the VMR-WB Interoperable mode to AMR-WB at 12.65, 8.85, or 6.60 kbps.

Depending on the speech signal characteristics in each frame and the selected operating mode, the built-in rate selection mechanism chooses a particular and permissible encoding type operating at one of the bit rates available in CDMA Rate Set II. These rates are Full Rate (FR) at 13.3 kbps, Half Rate (HR) at 6.2 kbps, Quarter-Rate (QR) at 2.7 kbps, and Eighth Rate (ER) at 1.0 kbps.

The analysis frame size is 20 ms. The encoding techniques utilized at FR or HR frames are based on the Algebraic CELP (ACELP) paradigm, while the encoding techniques used at QR or ER exploit Linear Prediction (LP) synthesis filter excited by a random noise with appropriately scaled energy. The encoding types are summarized in Table 4.1-1.

The encoder flow chart is shown in Figure 4.1-2. The pre-processing functions comprise sampling conversion, high-pass filtering, and spectral pre-emphasis. Spectral analysis is done twice per frame and provides the energy per critical bands. The critical band energies are used for the Voice Activity Detection (VAD) and the Noise Reduction.

Table 4.1-1: VMR-WB encoding types and their brief description

Encoding Types	Brief Description
Generic FR	General purpose FR codec
12.65 kbps Interoperable FR	General purpose FR codec interoperable with AMR-WB @ 12.65 kbps
8.85 kbps Interoperable FR	General purpose FR codec interoperable with AMR-WB @ 8.85 kbps
6.60 kbps Interoperable FR	General purpose FR codec interoperable with AMR-WB @ 6.60 kbps
Signaling HR	General purpose HR codec used for packet level signaling
12.65 kbps Interoperable HR	General purpose HR codec used for signaling in the 12.65 kbps AMR-WB interoperable mode
8.85 kbps Interoperable HR	General purpose HR codec used for signaling in the 8.85 kbps AMR-WB interoperable mode
6.60 kbps Interoperable HR	General purpose HR codec used for signaling in the 6.60 kbps AMR-WB interoperable mode
Generic HR	General purpose HR codec
Voiced HR	Voiced frame encoding at HR
Unvoiced HR	Unvoiced frame encoding at HR
Unvoiced QR	Unvoiced frame encoding at QR
CNG QR	Comfort noise generator for the AMR-WB interoperable mode at QR
CNG ER	Comfort noise generator at ER

The LP analysis is done similar to the AMR-WB standard. The open-loop pitch value is searched three times per frame using a pitch-tracking algorithm. The signal modification function modifies the original signal to make the encoding easier for the HR voiced encoder. It also contains an inherent classifier for classification of those frames that are suitable for HR voiced encoding. The other encoding techniques are determined in the rate selection block.

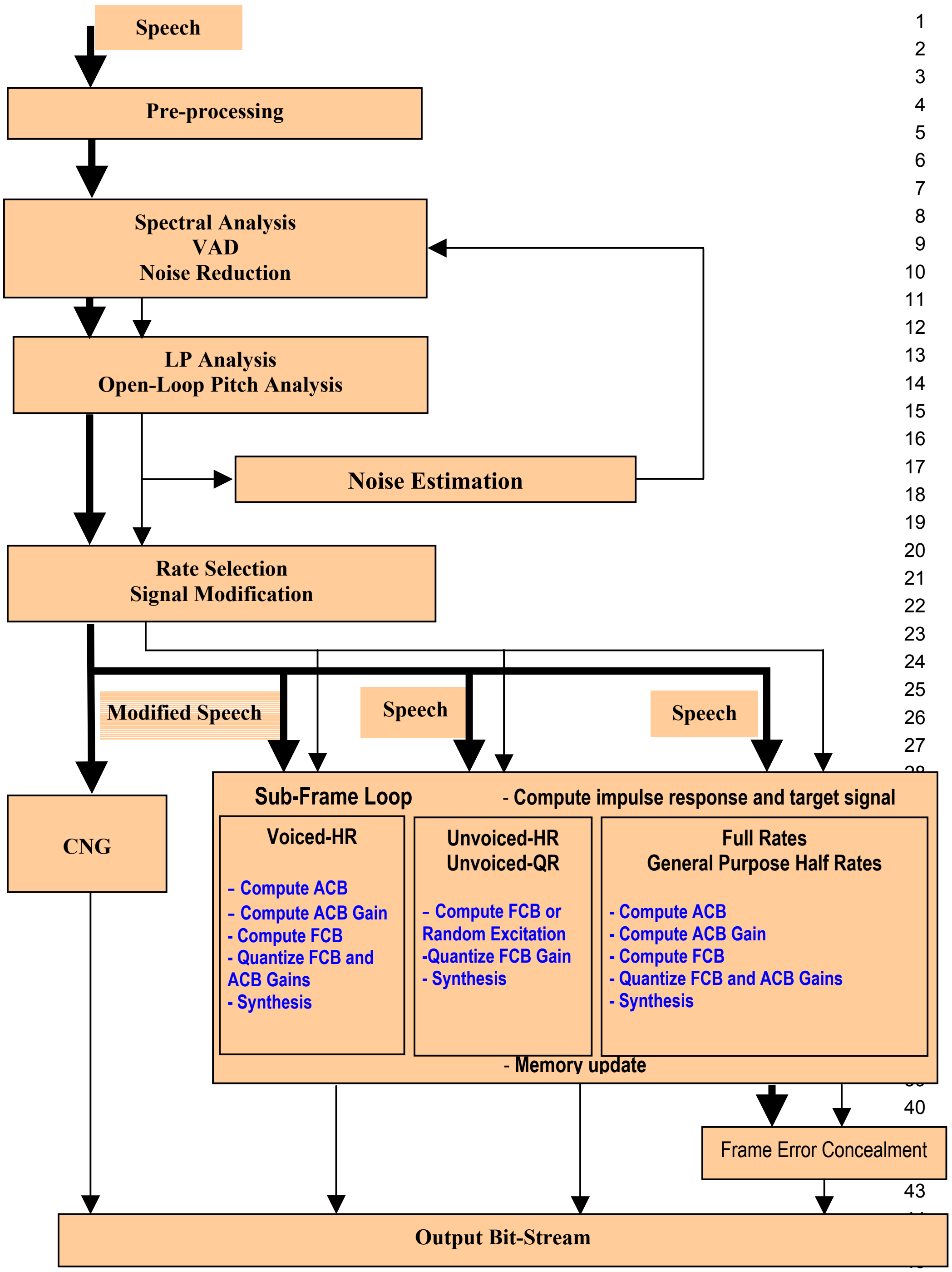

Figure 4.1-2: The VMR-WB Encoder Flow Chart

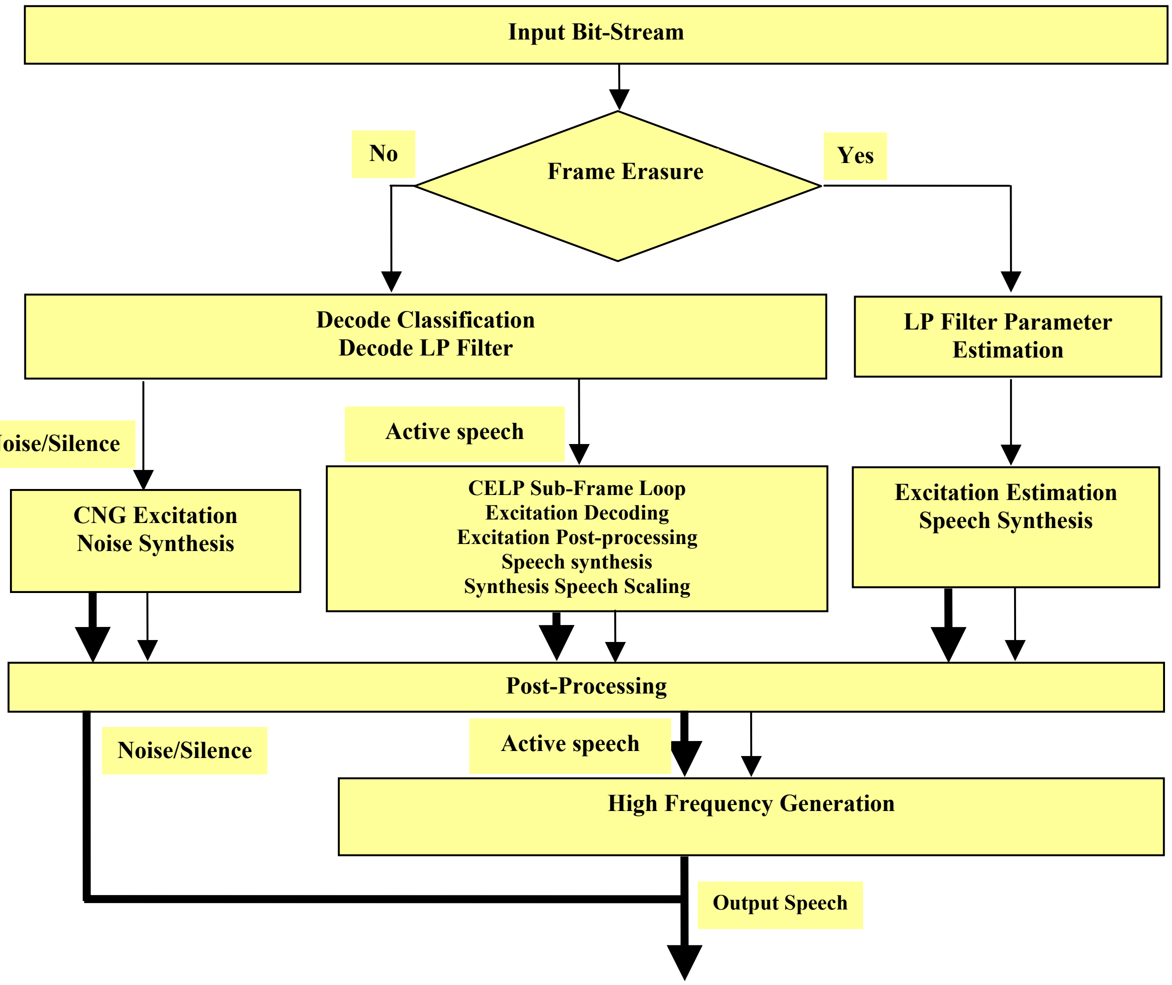

Figure 4.1-3: The VMR-WB Decoder Flow Chart

The active speech is then processed through the CELP sub-frame loop using an appropriate coding technique. If the Generic FR coder is selected, supplementary information is added for better frame error concealment and recovery upon frame erasure detection at the decoder. The Comfort Noise Generation (CNG) module using CNG QR or CNG ER, depending whether VMR-WB is engaged in an AMR-WB interoperable call, encodes the inactive speech.

The decoder flow chart is shown in Figure 4.1-3. The erased frames following an active speech period are processed separately independent of mode or rate. The silence or noise-only frames are generated using the CNG decoder. The active speech frames are processed with conventional CELP decoding. The post-processing consists of spectral de-emphasis, sampling conversion to the output frequency, and low-frequency enhancement. Finally, high frequencies are regenerated and added to active speech frames if the output is sampled at 16 kHz.

The VMR-WB algorithmic delay is a fixed delay. For wideband operation, the encoder algorithmic delay is 32.8125 ms accounting for the frame length of 20 ms, 11.875 ms of lookahead and 0.9375 ms of re-sampling filter delay. The decoder delay is 0.9375 ms, which comprises re-sampling filter delay. The total encoder-decoder delay for wideband operation is 33.75 ms. For narrowband operation, the encoder delay is 32.875 ms and the decoder delay is 2.1875 ms. The total encoder-decoder delay for narrowband operation is 35.0635 ms.

As an example, the following Tables provide rate usage of VMR-WB modes and their corresponding average data rates under clean and background noise conditions.

Table 4.1-2: Rate Percentages and Average Data Rates for Clean Speech at Nominal Level

	Mode 0	Mode 1	Mode 2	Mode 3
Rate 1	56.17%	36.28%	20.51%	60.11%
Rate ½	3.95%	23.83%	31.22%	0
Rate ¼	0	0	8.38%	6.61%
Rate 1/8	39.89%	39.89%	39.89%	33.28%
ADR kbps	9.0903	7.6584	6.2211	9.4934

Table 4.1-3: Rate Percentages and Average Data Rates for Speech + Street Noise at 15 dB SNR

	Mode 0	Mode 1	Mode 2	Mode 3
Rate 1	54.57%	37.77%	22.55%	57.57%
Rate ½	2.99%	19.80%	24.52%	0
Rate ¼	0	0	10.49%	7.32%
Rate 1/8	42.43%	42.43%	42.43%	35.11%
ADR kbps	8.8379	7.6279	6.1545	9.1850

Table 4.1-2 provides the percentages of the selected rates and the ADR for a clean sample speech database at nominal level (-22 dB) that was used for speech quality tests during the VMR-WB selection. The voice activity for this database is about 63%; hence, that database does not represent typical conversational speech traffic.

Table 4.1-3 provides the percentages of the selected rates and the ADR for speech + street noise at 15 dB signal-to-noise ratio (SNR), which was generated from the same database.

4.2 Bit Allocation Tables

The general bit allocation for various encoding schemes used in the VMR-WB codec are given in Table 4.2-1.

Table 4.2-1: Bit allocation for all encoding schemes used in the VMR-WB codec

Parameter	Generic FR	12.65 kbps Interoperable FR	8.85 kbps Interoperable FR	6.6 kbps Interoperable FR	Signaling HR	Unvoiced HR	Generic HR
Class Info/Frame Identifier	*	13	13	13	3*	2	1
VAD Flag	0	1	1	1	0	0	0
LP Parameters	46	46	46	36	46	46	36
Pitch Delay	30	30	26	23	30	0	13
Pitch Filtering	4	4	0	0	4	0	0
Gains	28	28	24	24	28	24	26
Fixed- Codebook	144	144	80	48	0	52	48
FER protection bits	14*	0	0	0	8*	0	0
Unused bits	0		76	121	5	0	0
Total	266	266	266	266	124	124	124
Parameter	**Voiced HR**	**12.65 kbps Interoperable HR**	**8.85 kbps Interoperable HR**	**6.6 kbps Interoperable HR**	**Unvoiced QR**	**CNG-QR**	**CNG-ER**
Class Info/Frame Identifier	3	13	13	13	1	1	0
VAD Flag	0	1	1	1	0	0	0
LP Parameters	36	46	46	36	32	28	14
Pitch Delay	9	30	26	23	0	0	0
Pitch Filtering	2	4	0	0	0	0	0
Gains	26	28	24	24	20	6	6
Fixed- Codebook	48	2	14	27	0	0	0
FER protection bits	0	0	0	0	0	0	0
Unused bits	0	0	0	0	1	19	0
Total	124	124	124	124	54	54	20

*Note: In Generic FR and Signaling HR encoding types, FER protection bits are instead used to transmit signal energy. However, only 63 out of 64 possible combinations are used for energy encoding. The bit combination ‘111110’ (i.e., decimal 62) is reserved and used to distinguish between different interoperable encoding types (This bit combination is part of the Class Info/Frame Identifier bits in the interoperable encoding schemes). More details are provided in Section 8.

The detailed bit allocation for various VMR-WB encoding types are given in Table 4.2-2 to Table 4.2-5.

Table 4.2-2: Detailed bit allocation for Full-Rate encoding schemes used in VMR-WB

	VMR-WB Encoding Type (4 sub-frames/frame)			
Parameter	**Generic Full-Rate**	**12.65 Interoperable Full-Rate**	**8.85 Interoperable Full-Rate**	**6.60 Interoperable Full-Rate**
VAD Flag	-	1 bit/Frame	1 bit/Frame	1 bit/Frame
ISFs	2 stage MA Prediction Split-VQ 1st stage: 8+8 bits/frame 2nd stage: 6+7+7+5+5 bits/frame	2 stage MA Prediction Split-VQ 1st stage: 8+8 bits/frame 2nd stage: 6+7+7+5+5 bits/frame	2 stage MA Prediction Split-VQ 1st stage: 8+8 bits/frame 2nd stage: 6+7+7+5+5 bits/frame	2 stage MA Prediction Split-VQ 1st stage: 8+8 bits/frame 2nd stage: 7+7+6 bits/frame
Fixed-Codebook Indices	36 bits/sub-frame	36 bits/sub-frame	20 bits/sub-frame	12 bits/sub-frame

Adaptive Codebook	9+6+9+6 bits/frame	9+6+9+6 bits/frame	8+5+8+5 bits/frame	8+5+5+5 bits/frame
Codebook Gains	Joint VQ 7 bits/sub-frame	Joint VQ 7 bits/sub-frame	Joint VQ 6 bits/sub-frame	Joint VQ 6 bits/sub-frame
Pitch Filtering	1 bit/sub-frame	1 bit/sub-frame	-	-
FER Protection	2 bits/frame for frame classification 6 bits/frame for first glottal pulse position 6 bits/frame for synthesized speech energy	-	-	-
Total Bits/Frame	266	266	266	266

Table 4.2-3: Detailed bit allocation for Half-Rate encoding schemes used in VMR-WB

	VMR-WB Encoding Type (4 sub-frames/frame)						
Parameter	**Generic Half-Rate**	**12.65 Interoperable Half-Rate**	**8.85 Interoperable Half-Rate**	**6.60 Interoperable Half-Rate**	**Signaling Half-Rate**	**Voiced Half-Rate**	**Unvoiced Half-Rate**
Class Info	1 bit/frame	3 bits/frame	3 bits/frame	3 bits/frame	3 bits/frame	3 bits/frame	2 bits/frame
VAD Flag	0	1 bit/Frame	1 bit/Frame	1 bit/Frame	0	0	0
ISFs	2 stage MA Prediction Split-VQ 1^{st} stage: 8+8 bits/frame 2^{nd} stage: 7+7+6 bits/frame	2 stage MA Prediction Split-VQ 1^{st} stage: 8+8 bits/frame 2^{nd} stage: 6+7+7+5+5 bits/frame	2 stage MA Prediction Split-VQ 1^{st} stage: 8+8 bits/frame 2^{nd} stage: 6+7+7+5+5 bits/frame	2 stage MA Prediction Split-VQ 1^{st} stage: 8+8 bits/frame 2^{nd} stage: 7+7+6 bits/frame	2 stage MA Prediction Split-VQ 1^{st} stage: 8+8 bits/frame 2^{nd} stage: 6+7+7+5+5 bits/frame	2 stage AR Prediction Split-VQ 1^{st} stage: 8+8 bits/frame 2^{nd} stage: 7+7+6 bits/frame	2 stage MA Prediction Split-VQ 1^{st} stage: 8+8 bits/frame 2^{nd} stage: 6+7+7+5+5 bits/frame
Fixed-Codebook Indices	12 bits/sub-frame	0	0	0	0	12 bits/sub-frame	Gaussian Codebook 13 bits/sub-frame
Adaptive Codebook	8+5 bits/frame	9+6+9+6 bits/frame	8+5+8+5 bits/frame	8+5+5+5 bits/frame	9+6+9+6 bits/frame	9 bits/frame	0
Gain Quantization Flag	1+1 bits/frame	0	0	0	0	1+1 bits/frame	0
Codebook Gains	Joint VQ 6 bits/sub-frame	Joint VQ 7 bits/sub-frame	Joint VQ 6 bits/sub-frame	Joint VQ 6 bits/sub-frame	Joint VQ 7 bits/sub-frame	Joint VQ 6 bits/sub-frame	Scalar Quantization 6 bits/sub-frame
Pitch Filtering	1+1 bits/frame	1 bit/sub-frame	0	0	1 bit/sub-frame	1+1 bits/frame	0
FER Protection	0	0	0	0	2 bits/frame for frame classification 6 bits/frame for synthesized speech energy	0	0
Total Bits/Frame	124	124	124	124	124	124	124

Table 4.2-4: Detailed bit allocation for Quarter-Rate encoding schemes used in VMR-WB

Parameter	VMR-WB Encoding Type	
	CNG Quarter-Rate (1 sub-frame/frame)	Unvoiced Quarter-Rate (4 sub-frames/frame)
Class Info	0	2 bits/frame
ISFs	Single stage Split-VQ 6+6+6+5+5 bits/frame	2 stage MA Prediction Split-VQ 1st stage: 8+8 bits/frame 2nd stage: 5+5+6 bits/frame
Excitation Gain	6 bits/frame	5 bits/sub-frame
Total Bits/Frame	54	54

Table 4.2-5: Detailed bit allocation for Eighth-Rate encoding scheme used in VMR-WB

Parameter	VMR-WB Encoding Type (1 sub-frame/frame)
	CNG Eighth-Rate
ISFs	Single stage Split-VQ 5+5+4 bits/frame
Excitation Gain	6 bits/frame
Total Bits/Frame	20

4.3 VMR-WB Symbol Table

This specification uses the following symbols:

Symbol	Description
$A(z)$	The inverse filter with unquantized coefficients
$\hat{A}(z)$	The inverse filter with quantized coefficients
$H(z)=\frac{1}{\hat{A}(z)}$	The speech synthesis filter with quantized coefficients
a_i	The unquantized linear prediction parameters (direct form coefficients)
$\hat{a}_i$	The quantified linear prediction parameters
m	The order of the LP model
$W(z)$	The perceptual weighting filter (unquantized coefficients)
γ_1	The perceptual weighting factor
T	The integer pitch lag nearest to the closed-loop fractional pitch lag of the subframe
β	The adaptive pre-filter coefficient (the quantified pitch gain)
$H_{h1}(z)$	Preprocessing high-pass filter
$w(n)$	LP analysis window
L_1	Length of the first part of the LP analysis window $w(n)$
L_2	Length of the second part of the LP analysis window $w(n)$
$r_c(k)$	The autocorrelations of the windowed speech $s'(n)$
$w_{lag}(i)$	Lag window for the autocorrelations (60 Hz bandwidth expansion)
f_0	The bandwidth expansion in Hz
f_s	The sampling frequency in Hz
$r'(k)$	The modified (bandwidth expanded) autocorrelations
$E(i)$	The prediction error in the ith iteration of the Levinson algorithm
k_i	The ith reflection coefficient

$a_j^{(i)}$	The *j*th direct form coefficient in the *i*th iteration of the Levinson algorithm
$F_1'(z)$	Symmetric ISF polynomial
$F_2'(z)$	Asymmetric ISF polynomial
$F_1(z)$	Polynomial $F_1'(z)$
$F_2(z)$	Polynomial $F_2'(z)$ with roots z = 1 and z = –1 eliminated
q_i	The Immitance spectral pairs (ISFs) in the cosine domain
$\mathbf{q}$	An ISF vector in the cosine domain
$\hat{\mathbf{q}}_i^{(n)}$	The quantified ISF vector at the *i*th subframe of the frame *n*
ω_i	The Immitance spectral frequencies (ISFs)
$T_m(x)$	A *m*th order Chebyshev polynomial
$f_1(i)$, $f_2(i)$	The coefficients of the polynomials $F_1(z)$ and $F_2(z)$
$f_1'(i), f_2'(i)$	The coefficients of the polynomials $F_1'(z)$ and $F_2'(z)$
$f(i)$	The coefficients of either $F_1(z)$ or $F_2(z)$
C(x)	Sum polynomial of the Chebyshev polynomials
x	Cosine of angular frequency ω
λ_k	Recursion coefficients for the Chebyshev polynomial evaluation
f_i	The Immitance spectral frequencies (ISFs) in Hz
$\mathbf{f}^t = [f_1\, f_2 \ldots f_{16}]$	The vector representation of the ISFs in Hz
$\mathbf{z}(n)$	The mean-removed ISF vector at frame *n*
$\mathbf{r}(n)$	The ISF prediction residual vector at frame *n*
$\mathbf{p}(n)$	The predicted ISF vector at frame *n*
$\hat{\mathbf{r}}(n-1)$	The quantified residual vector at the past frame
$\hat{\mathbf{r}}_i^k$	The quantified ISF subvector i at quantization index *k*
d_i	The distance between the Immitance spectral frequencies f_{i+1} and f_{i-1}
$h(n)$	The impulse response of the weighted synthesis filter
$H(z)W(z)$	The weighted synthesis filter
T_1	The integer nearest to the fractional pitch lag of the previous (1st or 3rd) subframe
$s'(n)$	The windowed speech signal
$s_w(n)$	The weighted speech signal
$\hat{s}(n)$	Reconstructed speech signal
$x(n)$	The target signal for adaptive codebook search
$x_2(n), \mathrm{x}_2^t$	The target signal for algebraic codebook search
$r(n)$	The LP residual signal
$c(n)$	The fixed-codebook vector
$v(n)$	The adaptive codebook vector
$y(n) = v(n)*h(n)$	The filtered adaptive codebook vector
$y_k(n)$	The past filtered excitation
$u(n)$	The excitation signal
$\hat{u}'(n)$	The gain-scaled emphasized excitation signal
T_{op}	The best open-loop lag
t_{min}	Minimum lag search value
t_{max}	Maximum lag search value
$R(k)$	Correlation term to be maximized in the adaptive codebook search
$R(k)_t$	The interpolated value of $R(k)$ for the integer delay *k* and fraction *t*
A_k	Correlation term to be maximized in the algebraic codebook search at index *k*

C_k	The correlation in the numerator of A_k at index k
E_{Dk}	The energy in the denominator of A_k at index k
$\mathbf{d} = \mathbf{H}^t\,\mathbf{x}_2$	The correlation between the target signal $x_2(n)$ and the impulse response $h(n)$, i.e. backward filtered target
$\mathbf{H}$	The lower triangular Toeplitz convolution matrix with diagonal $h(0)$ and lower diagonals $h(1)$, ..., $h(63)$
$\Phi = \mathbf{H}^t\mathbf{H}$	The matrix of correlations of $h(n)$
$d(n)$	The elements of the vector **d**
$\phi(i, j)$	The elements of the symmetric matrix Φ
c_k	The innovation vector
C	The correlation in the numerator of A_k
m_i	The position of the ith pulse
ϑ_i	The amplitude of the ith pulse
N_p	The number of pulses in the fixed-codebook excitation
E_D	The energy in the denominator of A_k
$res_{LTP}(n)$	The normalized long-term prediction residual
$b(n)$	The signal used for presetting the signs in algebraic codebook search
$s_b(n)$	The sign signal for the algebraic codebook search
$d'(n)$	Sign extended backward filtered target
$\phi'(i, j)$	The modified elements of the matrix Φ, including sign information
$\mathbf{z}^t$, $z(n)$	The fixed-codebook vector convolved with $h(n)$
$E(n)$	The mean-removed innovation energy (in dB)
$\overline{E}$	The mean of the innovation energy
$\widetilde{E}(n)$	The predicted energy
$[b_1\ b_2\ b_3\ b_4]$	The MA prediction coefficients
$\hat{R}(k)$	The quantified prediction error at subframe k
E_I	The mean innovation energy
$R(n)$	The prediction error of the fixed-codebook gain quantization
E_Q	The quantization error of the fixed-codebook gain quantization
$e(n)$	The states of the synthesis filter $1/\hat{A}(z)$
$e_w(n)$	The perceptually weighted error of the analysis-by-synthesis search
η	The gain scaling factor for the emphasized excitation
g_c	The fixed-codebook gain
g'_c	The predicted fixed-codebook gain
$\hat{g}_c$	The quantified fixed-codebook gain
g_p	The adaptive codebook gain
$\hat{g}_p$	The quantified adaptive codebook gain
$\gamma_{gc} = g_c / g'_c$	A correction factor between the gain g_c and the estimated one g'_c
$\hat{\gamma}_{gc}$	The optimum value for γ_{gc}
γ_{sc}	Gain scaling factor
s'(n)	Preprocessed signal
s(n)	Denoised signal
$x_w(n)$	Windowed signal for spectral analysis
$X^{(1)}(k)$	FFT of $x_w(n)$ (first spectral analysis)
$X^{(2)}(k)$	FFT of $x_w(n)$ (second spectral analysis)
$X^{(0)}(k)$	Second spectral analysis in the past frame

$E_{CB}^{(j)}(i)$	Average energy per critical band from spectral analysis j (1 or 2)
$E_{BIN}^{(j)}(i)$	Energy per bin from spectral analysis j (1 or 2)
E_t	Total frame energy in dB
$E_{av}(i)$	Average energy per critical band in a speech frame
$SNR_{CB}(i)$	SNR per critical band
$N_{CB}(i)$	Estimated noise energy per critical band
SNR_{LT}	Long term SNR
N_{tot}	Total noise energy per frame in dB
$\overline{E}_f$	Long term average frame energy
$\overline{N}_f$	Long term average noise energy
$\overline{E}_{CB}(i)$	Frame energy per critical band
$N_{tmp}(i)$	Temporary updated noise energy
r_e	Noise correction factor
NR_{max}	Maximum allowed noise reduction in dB (default 14 dB)
g_{dB}	Maximum allowed noise attenuation level (in linear scale)
g_s	Scaling gain per critical band
$X'_R(k)$ and $X'_I(k)$	Scaled spectrum
K_{voic}	Number of voiced critical bands
$g_{CB,LP}$	Smoothed scaling gain used for noise reduction per critical band
$g_{BIN,LP}$	Smoothed scaling gain used for noise reduction per bin
$M_{CB}(i)$	Number of bins per critical band i
$\tilde{d}(n)$	Delay contour
$\breve{r}(n)$	Modified residual
$\overline{d}_k$	Pitch delay at the end of present frame in signal modification
$\overline{d}_{k-1}$	Pitch delay at the end of previous frame in signal modification

4.4 Abbreviations

This specification uses the following abbreviations:

Abbreviation	Description
3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnership Project 2
AMR-WB	Adaptive Multi-Rate Wideband
AR	Auto Regressive
CNG	Comfort Noise Generation
ER	Eighth-Rate
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FR	Full-Rate
HR	Half-Rate
IIR	Infinite Impulse Response
ISP	Immitance Spectral Pairs

ITU-T	International Telecommunication Union/Telecommunication Standardization Sector
LP	Linear Predictive Coding
LPC	Linear Prediction
LTP	Long-Term Prediction
MA	Moving Average
MIME	Multi-Purpose Internet Mail Extension
Modified-IRS	Modified Intermediate Response System
QR	Quarter-Rate
SID	Silence Descriptor
SNR	Signal to Noise Ratio
VAD	Voice Activity Detection
VMR-WB	Variable-Rate Multimode Wideband

5 FUNCTIONAL DESCRIPTION OF THE VMR-WB ENCODER

The VMR-WB encoder operates on 20 ms frames. The encoding procedure consists of the following:

- Preprocessing which consists of sampling conversion to 12800 samples/second, high-pass filtering and pre-emphasis.
- Spectral analysis which is used for voice activity detection and noise reduction
- Detection of narrow-band inputs
- Voice activity detection
- Noise estimation
- Noise reduction
- Linear prediction analysis, LP to ISF conversion, and interpolation.
- Computation of weighted speech signal
- Open-loop pitch analysis
- Background noise update
- Rate selection algorithm
- Signal modification and refinement of rate selection decision
- Frame encoding using the selected rate and encoding type (e.g. generic FR, voiced HR, unvoiced HR, CNG-QR, CNG-ER, etc.)

The details of the above listed encoding procedure will be described in the following sections.

5.1 Pre-Processing

5.1.1 Sampling Conversion

Routine Name: `modify_Fs`
Inputs:

- S_8 (n) or S_{16} (n): The input speech signal sampled at 8 or 16 kHz depending on the operational bandwidth, respectively.

Outputs:

- $s_{12.8}(n)$: The re-sampled and filtered speech signal.

Initialization:

- The FIR filter memory is set to all zeros at initialization.

The linear predictive (LP) analysis, long-term prediction (LTP), and computation of fixed-codebook parameters are performed by the VMR-WB encoder at 12.8 kHz sampling rate. Therefore, the wideband input signal is decimated from 16 to 12.8 kHz. The sampling conversion is performed by first up-sampling by 4, then filtering the output through low-pass FIR filter $H_{dec,16}(z)$ that has the cut off frequency at 6.4 kHz. Then, the signal is down-sampled by 5. The filtering delay is 15 samples at 16 kHz sampling frequency.

If the input signal sampled at 16 kHz is denoted $s_{16}(n)$ then sampling conversion is performed as follows using a 121-tap FIR filter $H_{dec,16}(z)$. First, the signal is up sampled to 64 kHz by inserting three zero-valued samples between each 2 samples for each 20ms frame of 320 samples at 16 kHz sampling

$$s_{64}(n) = \begin{cases} s_{16}(n/4), & \text{if } n/4 = \lfloor n/4 \rfloor \\ 0, & \text{otherwise} \end{cases} \quad 0 \le n < 1280 \qquad (5.1.1\text{-}1)$$

where $s_{16}(n)$ is the signal at 16 kHz sampling and $s_{64}(n)$ is the signal at 64 kHz sampling. Then, the signal $s_{64}(n)$ is filtered through the filter $H_{dec,16}(z)$ and decimated by 5 by keeping one out of 5 samples. The filter $H_{dec,16}(z)$ is a 121-tap linear phase FIR filter having a cut-off frequency at 6.4 kHz in the 64 kHz in the up-sampled domain. The filtering and decimation can be done using the relation

$$s_{12.8}(n) = \sum_{i=-60}^{60} s_{64}(5n+i)h_{16}(i), \qquad n = 0,...,255 \tag{5.1.1-2}$$

The operations in Equations (5.1.1-1) and (5.1.1-2) can be implemented in one step by using only fourth of the filter coefficients at a time with an initial phase related to the sampling instant *n*. That is

$$s_{12.8}(n) = \sum_{i=-K+1}^{K} s_{16}(m+i)h_{16}(4i-f), \qquad n = 0,...,255 \tag{5.1.1-3}$$

where, $m=\left\lfloor \frac{5}{4}n \right\rfloor$ and *K*=60/4=15. The initial phase f for the re-sampling ratio 4/5 is given by. $f = 5n - 4\left\lfloor \frac{5}{4}n \right\rfloor = 5n - 4m$ (In this case $f = n \bmod 4$).

The VMR-WB codec can also process narrowband input speech sampled at 8 kHz. In this case and assuming that the input speech signal has a modified-IRS spectral characteristics, the input speech is first filtered to compensate for the modified-IRS characteristics using a filter with the following transfer function. Note that the VMR-WB codec has been designed to operate internally on FLAT spectrum inputs.

$$H_{IRS\,\mathrm{mod}}(z) = \frac{1-0.5z^{-1}}{(1-f_{IRS\,\mathrm{mod}}z^{-1})}\ , \tag{5.1.1-4}$$

The constant f_{IRSmod} is set to 0.95 by default. If the narrowband input speech characteristics is not modified-IRS, the optimal performance of the codec for narrowband inputs can be tuned by changing this constant in a range between 0.95 and 0.5 corresponding to modified-IRS and FLAT narrowband input, respectively. The sampling conversion then consists of up-sampling from 8 kHz to 12.8 kHz. This is performed by first up-sampling by 8, then filtering the output at 64 kHz sampling rate through a low-pass FIR filter $H_{dec,8}(z)$ that has the cut off frequency at 4 kHz. Then, the signal is down-sampled by 5. A linear phase 129-tap FIR filter is used. The filtering delay is 64 samples at the 64 kHz frequency, which is equivalent to 8 samples at 8 kHz sampling frequency.

The filtering is performed similar to Equation (5.1.1-3) but with the new re-sampling ratio 8/5. That is

$$s_{64}(n) = \begin{cases} s_8(n/8), & \text{if } n/8 = \lfloor n/8 \rfloor \\ 0, & \text{otherwise} \end{cases} \qquad 0 \le n < 1280$$

$$s_{12.8}(n) = \sum_{i=-64}^{64} s_{64}(5n+i)h_8(i), \qquad n = 0,...,255$$

$$s_{12.8}(n) = \sum_{i=-K+1}^{K} s_8(m+i)h_8(8i-f), \qquad n = 0,...,255 \tag{5.1.1-5}$$

where $m=\left\lfloor \frac{5}{8}n \right\rfloor$, *K*=64/8=16, and $s_8(n)$ is the preprocessed signal at 8 kHz sampling. The initial phase f for the re-sampling ratio 8/5 is given by $f=5n-8\left\lfloor \frac{5}{8}n \right\rfloor = 5n-8m$. The upper half of the coefficients of the filter $H_{dec,16}(z)$ (n=0 to 60) are given as

$h_{16}(n)$ ={0.999980, 0.934870, 0.754870, 0.501632, 0.231474, -0.000000,
-0.152337, -0.209502, -0.181536, -0.098630, 0.000000,
0.078607, 0.114831, 0.104252, 0.058760, -0.000000,
-0.049374, -0.073516, -0.067781, -0.038681, 0.000000,
0.033082, 0.049550, 0.045881, 0.026258, -0.000000,
-0.022499, -0.033672, -0.031122, -0.017761, 0.000000,
0.015088, 0.022452, 0.020614, 0.011674, -0.000000,
-0.009736, -0.014331, -0.012999, -0.007264, 0.000000,
0.005872, 0.008488, 0.007546, 0.004123, -0.000000,
-0.003163, -0.004431, -0.003804, -0.001997, 0.000000,
0.001388, 0.001829, 0.001459, 0.000702, -0.000000,
-0.000383, -0.000424, -0.000267, -0.000091, 0.000000};

with $h_{16}(-n)=h_{16}(n),\ n=1,\ldots,60.$

The upper half of the coefficients of the filter $H_{dec,8}(z)$ (n=0 to 64) are given as

$h_8(n)$ = {
0.625000, 0.608656, 0.561206, 0.487214, 0.393681, 0.289232, 0.183124, 0.084212,
0.000000, -0.064103, -0.105241, -0.123231, -0.120363, -0.100899, -0.070383, -0.034841,
0.000000, 0.029389, 0.050024, 0.060296, 0.060272, 0.051463, 0.036420, 0.018228,
0.000000, -0.015586, -0.026613, -0.032111, -0.032070, -0.027308, -0.019240, -0.009571,
0.000000, 0.008043, 0.013583, 0.016181, 0.015927, 0.013342, 0.009229, 0.004498,
0.000000, -0.003604, -0.005918, -0.006835, -0.006500, -0.005240, -0.003472, -0.001613,
0.000000, 0.001150, 0.001758, 0.001870, 0.001615, 0.001161, 0.000670, 0.000262,
0.000000, -0.000113, -0.000116, -0.000066, -0.000018, 0.000000, 0.000000, 0.00, 0.00};

with $h_8(-n)=h_8(n),\ n=1,\ldots,64.$

5.1.2 High-Pass Filtering and Pre-emphasis

<u>Routine Name</u>: `hp50, preemph`

Inputs:

- $S_{12.8}(n)$: The re-sampled speech signal at 12.8 kHz.

Outputs:

- $s'(n)$: The high-pass filtered and pre-emphasized speech signal.

Initialization:

- The memory of $H_{hp}(z)$ and $H_{pre\text{-}emph}(z)$ filters are set to all zeros at initialization.

Following the sampling conversion stage, the re-sampled signal $s_{12.8}(n)$ is pre-processed to obtain the signal $s'(n)$. These pre-processing functions are applied to the signal prior to the encoding process: high-pass filtering and pre-emphasizing.

A high-pass filter is used to suppress undesired low frequency components of the input signal. The transfer function of the filter with cut off frequency of 50 Hz is given by

$$H_{hp}(z) = \frac{0.982910156 - 1.965820313z^{-1} + 0.982910156z^{-2}}{1 - 1.965820313z^{-1} + 0.966308593z^{-2}} \quad (5.1.2\text{-}1)$$

In the pre-emphasis, a first order high-pass filter is used to emphasize higher frequencies of the input speech and it is given by

$$H_{\text{pre-emph}}(z) = 1 - 0.68z^{-1} \quad (5.1.2\text{-}2)$$

5.2 Spectral Analysis

Routine Name: `analy_sp`
Inputs:

- $s'(n)$: The high-pass filtered and pre-emphasized speech signal,

Outputs:

- $X(k)$: 256-point FFT of the pre-processed input speech
- $E_{CB}(i)$: Average energy in ith critical band
- $E_{BIN}(k)$: Energy in the kth frequency bin
- E_t: Total frame energy

Initialization:

- None

Spectral analysis is used for the VAD, noise suppression, and signal classification functions.

The Discrete Fourier Transform is used to perform the spectral analysis and spectral energy estimation. The frequency analysis is done twice per frame using 256-point Fast Fourier Transform (FFT) with a 50 percent overlap. The positions of the analysis windows are appropriately selected so that all lookahead information is exploited. The beginning of the first window is placed 24 samples after the beginning of the current frame. The second window is placed 128 samples farther. A square root of a Hanning window (which is equivalent to a sine window) is used to weight the input signal for the frequency analysis. This window is particularly well suited for overlap-add methods. Thus, this particular spectral analysis is used in the noise suppression algorithm based on spectral subtraction and overlap-add analysis/synthesis. (Instead of using a Hanning window at the analysis stage and a rectangular window at synthesis stage, a square-root Hanning window is used at both stages). The square root Hanning window is given by

$$w_{FFT}(n) = \sqrt{0.5 - 0.5\cos\left(\frac{2\pi n}{L_{FFT}}\right)} = \sin\left(\frac{\pi n}{L_{FFT}}\right), \qquad n = 0,\ldots,L_{FFT}-1$$

(5.2-1)

where L_{FFT}=256 is the size of FFT analysis. Note that only half the window is computed and stored since it is symmetric (from 0 to $L_{FFT}/2$).

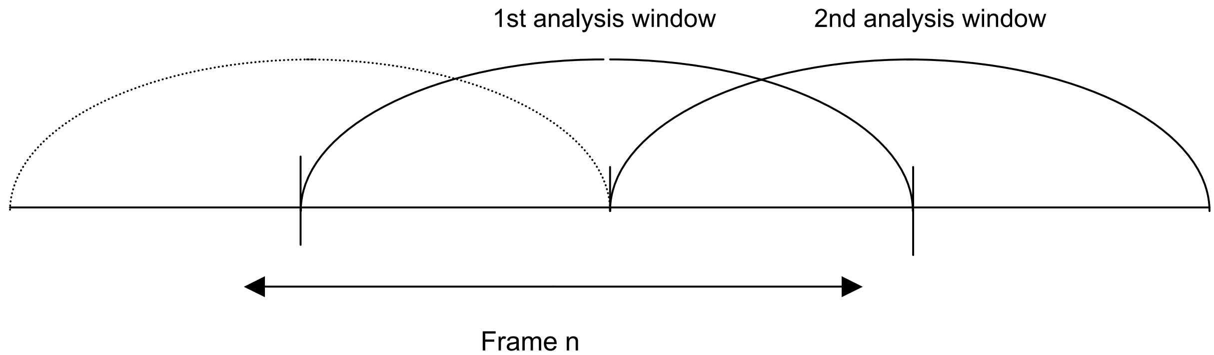

Figure 5.2-1: Relative positions of the spectral analysis windows

The windowed signal for both spectral analysis are obtained as

$$x_w^{(1)}(n) = w_{FFT}(n)s'(n+24), \qquad n = 0,\ldots,L_{FFT}-1$$
$$x_w^{(2)}(n) = w_{FFT}(n)s'(n+24+L_{FFT}/2), \qquad n = 0,\ldots,L_{FFT}-1$$

(5.2-2)

where s'(0) is the first sample in the present frame. The superscript (1) and (2) used to denote the first and the second frequency analysis, respectively, are dropped for simplicity up to Equation (5.2-5).

FFT is performed on both windowed signals to obtain two sets of spectral parameters per frame:

$$X(k) = \sum_{n=0}^{N-1} x_w(n) e^{-j2\pi\frac{kn}{N}}, \qquad k = 0,\ldots,L_{FFT}-1 \qquad (5.2\text{-}3)$$

The output of the FFT provides the real and imaginary parts of the spectrum denoted by $X_R(k)$, *k*=0 to 128, and $X_I(k)$, *k*=1 to 127. Note that $X_R(0)$ corresponds to the spectrum at 0 Hz (DC) and $X_R(128)$ corresponds to the spectrum at 6400 Hz. The spectrum at these points is only real valued and usually ignored in the subsequent analysis.

After FFT analysis, the resulting spectrum is divided into critical bands [19] using the intervals having the following upper limits (20 bands in the frequency range 0-6400 Hz): Critical bands = {100.0, 200.0, 300.0, 400.0, 510.0, 630.0, 770.0, 920.0, 1080.0, 1270.0, 1480.0, 1720.0, 2000.0, 2320.0, 2700.0, 3150.0, 3700.0, 4400.0, 5300.0, 6350.0} Hz.

The 256-point FFT results in a frequency resolution of 50 Hz (i.e., 6400/128). Thus after ignoring the DC component of the spectrum, the number of frequency bins per critical band is M_{CB}= {2, 2, 2, 2, 2, 2, 3, 3, 3, 4, 4, 5, 6, 6, 8, 9, 11, 14, 18, 21}, respectively.

The average energy in critical band is computed as

$$E_{CB}(i) = \frac{1}{(L_{FFT}/2)^2 M_{CB}(i)} \sum_{k=0}^{M_{CB}(i)-1} \left(X_R^2(k+j_i) + X_I^2(k+j_i) \right), \qquad i = 0,\ldots,19 \qquad (5.2\text{-}4)$$

where $X_R(k)$ and $X_I(k)$ are, respectively, the real and imaginary parts of the *k*th frequency bin and j_i is the index of the first bin in the *i*th critical band given by j_i ={1, 3, 5, 7, 9, 11, 13, 16, 19, 22, 26, 30, 35, 41, 47, 55, 64, 75, 89, 107}.

The spectral analysis module also computes the energy per frequency bin, E_{BIN} (*k*), for the first 17 critical bands (74 bins excluding the DC component)

$$E_{BIN}(k) = X_R^2(k) + X_I^2(k), \qquad k = 0,...,73 \tag{5.2-5}$$

Finally, the spectral analysis module computes the average total energy for both FFT analyses in a 20 ms frame by adding the average critical band energies E_{CB}. That is, the spectrum energy for a certain spectral analysis is computed as

$$E_{frame} = \sum_{i=0}^{19} E_{CB}(i) \tag{5.2-6}$$

and the total frame energy (in dB) is computed as the average of spectrum energies of both spectral analysis in a frame. That is

$$E_t = 10\log(0.5(E_{frame}(0) + E_{frame}(1)) \tag{5.2-7}$$

The output parameters of the spectral analysis module (both spectral analyses), that is average energy per critical band, the energy per frequency bin, and the total energy, are used in VAD, noise reduction, and rate selection modules.

Note that for narrowband inputs sampled at 8000 samples/second, after sampling conversion to 12800 samples/second, there is no content at both ends of the spectrum, thus the first lower frequency critical band as well as the last three high frequency bands are not considered in the computation of output parameters (only bands from *i*=1 to 16 are considered).

5.2.1 Detection of narrowband inputs

When processing wideband input speech sampled at 16 kHz and following the sampling conversion stage, the spectrum up to 6400 Hz is considered in the spectral analysis. Thus the minimum and maximum critical bands are set to b_{min}=0 and b_{max}=19. However, in case of narrowband inputs sampled at 8 kHz, after sampling conversion to 12.8 kHz, the maximum and minimum bands are set to b_{min}=1 and b_{max}=16. This implies that the spectrum below 150 Hz and above 3700 Hz is not considered in spectral analysis. For the cases where the input speech is over-sampled at 16 kHz while the input is actually a narrowband signal, a simple detection algorithm is applied to detect such signals and avoid processing them as wideband signals.

The detection is based on computing the smoothed energy in the upper two critical bands and comparing it to a certain threshold.

The mean energy in the upper two bands in a frame is given by

$$E_{18,19} = 0.25\left(E_{CB}^{(1)}(18) + E_{CB}^{(1)}(19) + E_{CB}^{(2)}(18) + E_{CB}^{(2)}(19)\right) \tag{5.2.1-1}$$

where $E_{CB}^{(1)}(i)$ and $E_{CB}^{(2)}(i)$ are the critical band energies from both spectral analysis. The smoothed energy in the upper two bands is then computed as

$$\overline{E}_{18,19} = 0.99\overline{E}_{18,19} + 0.01E_{18,19} \tag{5.2.1-2}$$

with initial value $\overline{E}_{18,19} = 2$.

The decision for the minimum and maximum used critical bands b_{min} and b_{max} is made a follows (using hysteresis):

If ($\overline{E}_{18,19} < 1$) then b_{min}=1 and b_{max}=16

Else If ($\overline{E}_{18,19} > 2$) then b_{min}=0 and b_{max}=19

This operation is performed only if the input signal is sampled at 16 kHz and if the mean energy in bands 15 and 16 (computed similar to Equation (5.2.1-1) is higher than 4.

5.3 Voice Activity Detection

Routine Name: `wb_vad`
Inputs:

- $E_{CB}(i)$: Average energy in ith critical band
- $N_{CB}(i)$: Noise estimate in ith critical band
- $\overline{E}_f$: Long-term average frame energy
- $\overline{N}_f$: Long-term average noise energy

Outputs:

- VAD_flag and local VAD flag

Initialization:

- $\overline{E}_f$ is initialized to 45 dB. $\overline{N}_f$ is initialized to total noise energy N_{tot}. N_{tot} initialization is provided in Section 5.4. VAD internal parameters of active speech counter and VAD hangover counter are initialized to 3 and 0, respectively.

The spectral analysis described in Section 5.2 is performed twice per frame. Let $E_{CB}^{(1)}(i)$ and $E_{CB}^{(2)}(i)$ denote the energy per critical band for the first and second spectral analysis, respectively (as computed in Equation (5.2-4)). The average energy per critical band for the whole frame and part of the previous frame is computed as

$$E_{av}(i) = 0.2E_{CB}^{(0)}(i) + 0.4E_{CB}^{(1)}(i) + 0.4E_{CB}^{(2)}(i) \tag{5.3-1}$$

where $E_{CB}^{(0)}(i)$ denote the energy per critical band from the second analysis of the previous frame. The signal-to-noise ratio (SNR) per critical band is then computed as

$$SNR_{CB}(i) = E_{av}(i) / N_{CB}(i) \quad \text{Constrained by } SNR_{CB} \geq 1. \tag{5.3-2}$$

where $N_{CB}(i)$ is the estimated noise energy per critical band as will be explained in Section 5.4.2. The average SNR per frame, in dB, is then computed as

$$SNR_{av} = 10\log\left(\sum_{i=b_{min}}^{b_{max}} SNR_{CB}(i)\right), \tag{5.3-3}$$

where b_{min}=0 and b_{max}=19 in case of wideband signals, and b_{min}=1 and b_{max}=16 in case of narrowband signals.

The voice activity is detected by comparing the average SNR per frame to a certain threshold, which is a function of the long-term SNR. The long-term SNR is given by

$$SNR_{LT} = \overline{E}_f - \overline{N}_f \qquad (5.3\text{-}4)$$

where $\overline{E}_f$ and $\overline{N}_f$ are computed using equations (5.4.3-1) and (5.4.3-2), respectively. The initial value of $\overline{E}_f$ is 45 dB.

The threshold is a piecewise linear function of the long-term SNR. Two thresholds are used, one for clean speech and one for noisy speech.

For wideband signals, if $SNR_{LT} < 35$ (noisy speech) then

$$th_{VAD} = 0.4346\, SNR_{LT} + 13.9575$$

Else (clean speech)

$$th_{VAD} = 1.0333\, SNR_{LT} - 7$$

For narrowband signals, if $SNR_{LT} < 29.6$ (noisy speech) then

$$th_{VAD} = 0.313\, SNR_{LT} + 14.6$$

Else (clean speech)

$$th_{VAD} = 1.0333\, SNR_{LT} - 7$$

Further, a hysteresis in the VAD decision is added to prevent frequent switching at the end of an active speech period. It is applied when the frame is in a soft hangover period or if the last frame is an active speech frame. The soft hangover period consists of the first 10 frames after each active speech interval longer than 2 consecutive frames. In case of noisy speech ($SNR_{LT} < 35$) the hysteresis decreases the VAD decision threshold by

$$th_{VAD} = 0.95 th_{VAD} \qquad (5.3\text{-}5)$$

In case of clean speech, the hysteresis decrements the VAD decision threshold by

$$th_{VAD} = th_{VAD} - 11 \qquad (5.3\text{-}6)$$

If the average SNR per frame is larger than the VAD decision threshold, that is, if $SNR_{av} > th_{VAD}$, then the frame is declared as an active speech frame and the VAD flag and a local VAD flag are set to 1. Otherwise the VAD flag and the local VAD flag are set to 0. However, in case of noisy speech, the VAD flag is forced to 1 in hard hangover frames, i.e. one or two inactive frames following a speech period longer than 2 consecutive frames. (The local VAD flag is then equal to 0 but the VAD flag is set to 1).

5.4 Primary Noise Parameter Estimation and Update

Routine Name: `noise_est_down, long_enr, correlation_shift`
Inputs:

- $E_{CB}(i)$: Average energy in *i*th critical band

- $N_{CB}(i)$: Noise estimate in *i*th critical band

Outputs:

- N_{tot}: Total noise energy
- E_{rel}: Relative frame energy
- $\overline{N}_f$: Long-term average noise energy
- $\overline{E}_f$: Long-term average frame energy
- $N_{CB}(i)$: Noise estimate in *i*th critical band
- r_e: Noise correction factor

Initialization:

- $N_{CB}(i)$ is initialized to 0.03 unless otherwise described in the following section. N_{tot} is computed as a 10log of the sum of $N_{CB}(i)$.

In this section, the total noise energy, relative frame energy, update of long-term average noise energy and long-term average frame energy, average energy per critical band, and a noise correction factor are computed. Furthermore, noise energy initialization and update are described.

5.4.1 Total noise and Relative Frame Energy Estimation

The total noise energy per frame is computed as follows:

$$N_{tot} = 10\log\left(\sum_{i=0}^{19} N_{CB}(i)\right) \quad (5.4.1\text{-}1)$$

where $N_{CB}(i)$ is the estimated noise energy per critical band.

The relative energy of the frame is calculated by the difference between the frame energy in dB and the long-term average energy. The relative frame energy is given by

$$E_{rel} = E_t - \overline{E}_f \quad (5.4.1\text{-}2)$$

where E_t is given in Equation (5.2-7) and $\overline{E}_f$ is given in Equation (5.4.3-1).

5.4.2 Frame Energy per Critical Band, Noise Initialization, and Noise Update

The frame energy per critical band for the entire frame is computed by averaging the energies from both spectral analyses in the frame. That is,

$$\overline{E}_{CB}(i) = 0.5E_{CB}^{(1)}(i) + 0.5E_{CB}^{(2)}(i) \quad (5.4.2\text{-}1)$$

The noise energy per critical band $N_{CB}(i)$ is usually initialized to 0.03. However, in the first 5 subframes, if the signal energy is not too high or if the signal does not have strong high frequency components, then the noise energy is initialized using the energy per critical band so that the noise reduction algorithm can efficiently function from the beginning of the process. Two high frequency ratios are computed, $r_{15,16}$, as the ratio between the average energy of critical bands 15 and 16 and the average energy in the first 10 bands (i.e., mean of both spectral analyses), and $r_{18,19}$, as the ratio between the average energy of critical bands 18 and 19 and the average energy in the first 10 bands.

In the first 5 frames, if $E_t < 49$ and $r_{15,16}$<2 and $r_{18,19}$<1.5 then for the first 3 frames, then

$$N_{CB}(i) = \overline{E}_{CB}(i), \qquad i = 0,...,19 \tag{5.4.2-2}$$

and for the following two frames $N_{CB}(i)$ is updated by

$$N_{CB}(i) = 0.33 N_{CB}(i) + 0.66\overline{E}_{CB}(i), \qquad i = 0,...,19 \tag{5.4.2-3}$$

For the following frames, only noise energy update is performed for the critical bands where the signal energy is less than background noise energy. First, the temporary updated noise energy is computed as

$$N_{tmp}(i) = 0.9 N_{CB}(i) + 0.1\left(0.25 E_{CB}^{(0)}(i) + 0.75\overline{E}_{CB}(i)\right) \tag{5.4.2-4}$$

where $E_{CB}^{(0)}(i)$ corresponds to the second spectral analysis from the previous frame. Then the critical bands with their energy less than the background noise energy are updated as

For i=0 to 19, if $N_{tmp}(i) < N_{CB}(i)$ then $N_{CB}(i) = N_{tmp}(i)$

A second level of noise update is performed later in Section 5.9 by setting $N_{CB}(i) = N_{tmp}(i)$, if the frame is declared as inactive speech frame. The noise energy update has been divided into two stages because the noise parameter update can be performed only during inactive speech frames (i.e., silence intervals) and all the necessary parameters for the speech activity detection are thereby needed. These parameters are, however, dependent on LP analysis and open-loop pitch analysis, which are performed on the denoised speech signal. For the noise reduction algorithm in order to have as accurate noise estimate as possible, the noise estimation is thus updated by lowering the noise estimate, if necessary, before the noise reduction is performed and later by increasing the noise estimate, if necessary during silence intervals. Therefore, the noise level update is not increased in the first stage and the update can be done independent of the speech activity.

5.4.3 Long-Term Average Noise Energy and Frame Energy Update

Either the long-term average noise energy or the long-term average frame energy is updated in every frame. In case of active speech frames (VAD_flag = 1), the long-term average frame energy is updated as follows:

$$\overline{E}_f = 0.99\overline{E}_f + 0.01E_t \tag{5.4.3-1}$$

with initial value $\overline{E}_f = 45dB$. In case of inactive speech frames (VAD flag = 0), the long-term average noise energy is updated by

$$\overline{N}_f = 0.99\overline{N}_f + 0.01N_{tot} \tag{5.4.3-2}$$

The initial value of $\overline{N}_f$ is set equal to N_{tot} for the first 4 frames. Furthermore, in the first 4 frames, the value of $\overline{E}_f$ is constrained by $\overline{E}_f \geq \overline{N}_{tot} + 10$.

5.4.4 Noise Correction Factor

A correction factor r_e is added to the normalized correlation in order to compensate for the decrease of normalized correlation in the presence of background noise. It has been found that the dependence between this decrease r_e and the total background noise energy in dB is approximately exponential and can be estimated using following equation:

$$r_e = 0.00024492 \quad e^{0.1596\,(N_{tot} - g_{dB})} - 0.022 \quad \text{constrained by } r_e \geq 0. \tag{5.4.4-1}$$

where g_{dB} is the maximum allowed noise attenuation level in dB, which is set to 14 dB by default. It should be noted that under normal operation of the noise reduction algorithm g_{dB} is sufficiently high and thereby r_e is practically zero. It is only relevant when the noise reduction is disabled or if the background noise level is significantly higher than the maximum attenuation level.

5.5 Noise Suppression

Routine Name: `noise_sup`
Inputs:

- $E_{CB}(i)$: Average energy per critical band
- $E_{BIN}(k)$: Energy per frequency bin
- $N_{CB}(i)$: Noise estimate in *i*th critical band
- $X(k)$: 256-point FFT of the pre-processed input speech
- VAD_flag and local VAD flag
- K_{voic}: Voiced critical bands

Outputs:

- $s(n)$: The denoised speech signal

Initialization:

- The noise suppression buffers are initially set to zero. The number of voiced critical bands is initialized to 0. The smoothed scaling gains $g_{BIN,LP}(k)$ and $g_{CB,LP}(i)$ are initially set to 1.

Noise reduction is performed in the spectral domain. The denoised signal is then reconstructed using the overlap and add method. The reduction is performed by scaling the spectrum in each critical band with a scaling gain constrained between g_{min} and 1, which is derived from the signal-to-noise ratio (SNR) in that critical band. For frequencies lower than a certain frequency, the processing is performed on frequency bin basis and not on critical band basis. Thus, a scaling gain is applied on every frequency bin derived from the SNR in that bin (i.e., the SNR is computed using the bin energy divided by the noise energy of the critical band including that bin). This feature preserves the energy at frequencies close to harmonics and further prevents distortion while strongly reducing the noise between the harmonics. This technique is only utilized for voiced speech and, given the frequency resolution of the frequency analysis used, for speech segments with relatively short pitch period. These are scenarios where the noise between harmonics is most perceptible.

The minimum scaling gain g_{min} is derived from the maximum allowed noise reduction in dB, g_{dB} The maximum allowed reduction is an input parameter to the noise suppression unit with default value of 14 dB. Thus minimum-scaling gain is given by

$$g_{\min} = 10^{-g_{dB}/20} \tag{5.5-1}$$

and it is equal to 0.19953 for the default value of 14 dB.

In case of inactive speech frames with VAD_flag=0, the same scaling is applied over the entire spectrum and is given by $g_s = 0.9 g_{\min}$ if noise suppression is activated (if g_{min} is lower than 1). That is, the scaled real and imaginary components of the spectrum are given by

$$X'_R(k) = g_s X_R(k), \quad k = 1,...,128, \quad \text{and} \quad X'_I(k) = g_s X_I(k), \quad k = 1,...,127. \tag{5.5-2}$$

Note that for narrowband input speech, the maximum value of index *k* in Equation (5.5-2) is set to 79 (up to 3950 Hz). For purpose of noise suppression, the input signal is considered as narrowband only if the input sampling frequency is equal to 8000 Hz; i.e. in this case, the narrowband detector decision described in Section 5.2.1 is disregarded.

For active speech frames, the scaling gain is computed related to the SNR per critical band or per bin for the first voiced bands. If $K_{VOIC} > 0$ then per bin noise suppression is performed on the first K_{VOIC} bands. Per band noise suppression is used on the rest of the bands. In case $K_{VOIC} = 0$ per band noise suppression is used on the entire spectrum. The value of K_{VOIC} is updated as described in Section 5.9.1. The maximum value of K_{VOIC} is 17; therefore, per bin processing can be applied only on the first 17 critical bands corresponding to a maximum frequency of 3700 Hz. The maximum number of bins for which per bin processing can be used is 74 (the number of bins in the first 17 bands). An exception is considered for hard hangover frames that will be described later in this Section.

The scaling gain in a certain critical band, or for a certain frequency bin, is computed as a function of SNR and given by

$$(g_s)^2 = k_s \, SNR + c_s, \text{ constrained by } g_{\min} \le g_s \le 1 \tag{5.5-3}$$

The values of k_s and c_s are determined such that $g_s = g_{\min}$ for *SNR* = 1, and $g_s = 1$ for *SNR* = 45. That is, for $SNR \le 1$, the scaling is limited to g_s and for $SNR \ge 45$; no noise suppression is performed in the given critical band ($g_s = 1$). Note that the *SNR* is expressed as a ratio of energies. Thus, considering these two end points, the values of k_s and c_s in Equation (5.5-3) are given by

$$k_s = (1 - g_{\min}^2)/44 \quad \text{and} \quad c_s = (45 g_{\min}^2 - 1)/44. \tag{5.5-4}$$

The variable SNR in Equation (5.5-3) is either the SNR per critical band, $SNR_{CB}(i)$, or the SNR per frequency bin, $SNR_{BIN}(k)$, depending on the type of processing.

The SNR per critical band, corresponding to the first spectral analysis of the frame, is computed as follows:

$$SNR_{CB}(i) = \frac{0.2 E_{CB}^{(0)}(i) + 0.6 E_{CB}^{(1)}(i) + 0.2 E_{CB}^{(2)}(i)}{N_{CB}(i)} \qquad i = 0,...,19 \tag{5.5-5}$$

and the SNR per critical band, corresponding to the second spectral analysis of the frame, is computed as follows:

$$SNR_{CB}(i) = \frac{0.4E_{CB}^{(1)}(i) + 0.6E_{CB}^{(2)}(i)}{N_{CB}(i)} \qquad i = 0,...,19 \tag{5.5-6}$$

where $E_{CB}^{(1)}(i)$ and $E_{CB}^{(2)}(i)$ denote the energy per critical band for the first and second spectral analysis, respectively (as computed in Equation (5.2-4)), $E_{CB}^{(0)}(i)$ denotes the energy per critical band from the second spectral analysis of the previous frame, and $N_{CB}(i)$ denotes the noise energy estimate per critical band (see Section 5.9 for the update of $N_{CB}(i)$).

The SNR per critical bin in the *i*th critical band, corresponding to the first spectral analysis of the frame, is computed as

$$SNR_{BIN}(k) = \frac{0.2E_{BIN}^{(0)}(k) + 0.6E_{BIN}^{(1)}(k) + 0.2E_{BIN}^{(2)}(k)}{N_{CB}(i)}, \qquad k = j_i,..., j_i + M_{CB}(i) - 1 \tag{5.5-7}$$

and for the second spectral analysis, the SNR is computed as

$$SNR_{BIN}(k) = \frac{0.4E_{BIN}^{(1)}(k) + 0.6E_{BIN}^{(2)}(k)}{N_{CB}(i)}, \qquad k = j_i,..., j_i + M_{CB}(i) - 1 \tag{5.5-8}$$

where $E_{BIN}^{(1)}(k)$ and $E_{BIN}^{(2)}(k)$ denote the energy per frequency bin for the first and second spectral analysis, respectively (as computed in Equation (5.2-5)), $E_{BIN}^{(0)}(k)$ denote the energy per frequency bin from the second spectral analysis of the previous frame, $N_{CB}(i)$ denote the noise energy estimate per critical band, j_i is the index of the first bin in the *i*th critical band and $M_{CB}(i)$ is the number of bins in critical band *i* defined in Section 5.2.

In case of per critical band processing for a frequency band with index *i*, after determining the scaling gain as in Equation (5.5-5), and using SNR as defined in Equations (5.5-5) or (5.5-6), the actual scaling is performed using a smoothed scaling gain updated in every frequency analysis as

$$g_{CB,LP}(i) = \alpha_{gs} g_{CB,LP}(i) + (1 - \alpha_{gs}) g_s \tag{5.5-9}$$

where the smoothing factor is inversely proportional to the gain and is given by $\alpha_{gs} = 1 - g_s$. That implies the smoothing is stronger for smaller gains g_s. The scaling in the critical band is performed as

$$\begin{aligned} X_R^{'}(k + j_i) &= g_{CB,LP}(i)\, X_R(k + j_i), \qquad \text{and} \\ X_I^{'}(k + j_i) &= g_{CB,LP}(i)\, X_I(k + j_i), \qquad k = 0,..., M_{CB}(i) - 1 \end{aligned}, \tag{5.5-10}$$

where j_i is the index of the first bin in the critical band *i* and $M_{CB}(i)$ is the number of bins in that critical band.

In case of per bin processing in a frequency band with index *i*, after determining the scaling gain as in Equation (5.5-3), and using SNR as defined in Equations (5.5-7) or (5.5-8), the actual scaling is performed using a smoothed scaling gain updated in every frequency analysis which is calculated as follows:

$$g_{BIN,LP}(k) = \alpha_{gs} g_{BIN,LP}(k) + (1-\alpha_{gs}) g_s \qquad (5.5\text{-}11)$$

where $\alpha_{gs} = 1 - g_s$ similar to Equation (5.5-9).

While temporal smoothing of the gains prevents audible energy oscillations, control of the smoothing using α_{gs} prevents distortion in high-SNR speech segments preceded by low-SNR speech frames, as it is the case for voiced onsets for example. The scaling in the *i*th critical band is performed as

$$\begin{aligned} X'_R(k+j_i) &= g_{BIN,LP}(k+j_i)\, X_R(k+j_i), \quad \text{and} \\ X'_I(k+j_i) &= g_{BIN,LP}(k+j_i)\, X_I(k+j_i), \qquad k = 0,\dots,M_{CB}(i)-1 \end{aligned}, \qquad (5.5\text{-}12)$$

where j_i is the index of the first bin in the critical band *i* and $M_{CB}(i)$ is the number of bins in that critical band. The smoothed scaling gains $g_{BIN,LP}(k)$ and $g_{CB,LP}(i)$ are initially set to 1. Each time an inactive speech frame is processed (VAD_flag=0), the values of the smoothed gains are reset to $g_{\min}$ as defined in Equation (5.5-1).

As mentioned above, if $K_{VOIC} > 0$ per bin, noise suppression is performed on the first K_{VOIC} frequency bands, and per band noise suppression is performed on the remaining frequency bands using the procedures described above. Note that in every spectral analysis, the smoothed scaling gains $g_{CB,LP}(i)$ are updated for all critical bands (even for voiced bands processed with per bin processing - in this case $g_{CB,LP}(i)$ is updated with an average of $g_{BIN,LP}(k)$ associated with the band *i*). Similarly, scaling gains $g_{BIN,LP}(k)$ are updated for all frequency bins in the first 17 bands (up to bin 74). For frequency bands processed with per band processing, they are updated by setting them equal to $g_{CB,LP}(i)$ in these 17 specific bands.

Note that for clean speech, noise suppression is not performed in active speech frames (VAD_flag=1). This is detected by finding the maximum noise energy in all critical bands, $\max(N_{CB}(i)),\ i = 0,\dots,19,$ and if this value is less or equal 15 then no noise suppression is performed.

As mentioned above, for inactive speech frames (VAD_flag=0), a scaling of 0.9 $g_{\min}$ is applied over the entire spectrum, which is equivalent to removing a constant noise floor. For VAD short-hangover frames (VAD_flag=1 and local_VAD=0), per band processing is applied to the first 10 bands as described above (corresponding to 1700 Hz), and for the rest of the spectrum, a constant noise floor is subtracted by scaling the rest of the spectrum by a constant value $g_{\min}$. This measure reduces significantly high frequency noise energy oscillations. For these bands above the 10th band, the smoothed scaling gains $g_{CB,LP}(i)$ are not reset but updated using Equation (5.5-9) with $g_s = g_{\min}$ and the per bin smoothed scaling gains $g_{BIN,LP}(k)$ are updated by setting them equal to $g_{CB,LP}(i)$ in the corresponding critical bands.

In case of processing of narrowband speech signals (up-sampled to 12800 Hz), the noise suppression is performed on the first 17 bands (up to 3700 Hz). For the remaining 5 frequency bins between 3700 Hz and 4000 Hz, the spectrum is scaled using the last scaling gain g_s at the bin at 3700 Hz. For the remaining of the spectrum (from 4000 Hz to 6400 Hz), the spectrum is zeroed.

5.5.1 Reconstruction of Denoised Signal

After calculation of the scaled spectral components, $X'_R(k)$ and $X'_I(k)$, inverse FFT of the scaled spectrum is taken to obtain the windowed denoised signal in the time domain.

$$x_{w,d}(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{j2\pi\frac{kn}{N}}, \qquad n = 0,\ldots,L_{FFT}-1 \tag{5.5.1-1}$$

This is repeated for both spectral analyses in the frame to obtain the denoised windowed signals $x^{(1)}_{w,d}(n)$ and $x^{(2)}_{w,d}(n)$. For every half frame, the signal is reconstructed using an overlap-add scheme for the overlapping portions of the analysis. Since a square root Hanning window is used on the original signal prior to spectral analysis, the same window is applied at the output of the inverse FFT prior to overlap-add operation. Thus, the doubled windowed denoised signal is given by

$$\begin{aligned} x^{(1)}_{ww,d}(n) &= w_{FFT}(n) x^{(1)}_{w,d}(n), \quad n = 0,\ldots,L_{FFT}-1 \\ x^{(2)}_{ww,d}(n) &= w_{FFT}(n) x^{(2)}_{w,d}(n), \quad n = 0,\ldots,L_{FFT}-1 \end{aligned} \tag{5.5.1-2}$$

For the first half of the analysis window, the overlap-add scheme for constructing the denoised signal is formulated as

$$s(n+24) = x^{(0)}_{ww,d}(n + L_{FFT}/2) + x^{(1)}_{ww,d}(n), \qquad n = 0,\ldots,L_{FFT}/2-1 \tag{5.5.1-3}$$

and for the second half of the analysis window, the overlap-add operation for constructing the denoised signal is as follows:

$$s(n+24+L_{FFT}/2) = x^{(1)}_{ww,d}(n + L_{FFT}/2) + x^{(2)}_{ww,d}(n), \qquad n = 0,\ldots,L_{FFT}/2-1 \tag{5.5.1-4}$$

where $x^{(0)}_{ww,d}(n)$ is the double windowed denoised signal from the second analysis in the previous frame.

Note that with the overlap-add scheme described above, the denoised signal can be reconstructed up to 24 samples from the lookahead in addition to the present frame. However, another 128 samples are still needed to complete the lookahead needed for LP analysis and open-loop pitch analysis. This part is temporary obtained by inverse windowing the second half of the denoised windowed signal $x^{(2)}_{w,d}(n)$ without performing overlap-add operation. That is

$$s(n+24+L_{FFT}) = x^{(2)}_{ww,d}(n + L_{FFT}/2) / w^2_{FFT}(n + L_{FFT}/2), \qquad n = 0,\ldots,L_{FFT}/2-1 \tag{5.5.1-5}$$

Note that this portion of the signal is properly re-computed in the next frame using the overlap-add method as described above. In the following sections, the denoised signal *s* (n) is used as the input signal.

5.6 Linear Prediction Analysis and ISP Conversion

<u>Routine Name</u>: `analy_lp`
Inputs:

- *s* (*n*): The denoised speech signal
- $q_i^{(n-1)}$: The immitance spectral pairs of previous frame

Outputs:

- a_i : LP filter coefficients
- $q_i^{(n)}$: The immitance spectral pairs for current frame
- *E*(*i*): LP residual energies

Initialization:

- $q_i^{(n-1)}$ are initialized such that corresponding ISFs are equi-distant.

Short-term prediction, or LP analysis is performed once per speech frame using the autocorrelation approach with 30 ms asymmetric windows. An overhead of 5 ms is used in the autocorrelation computation. The frame structure is depicted below.

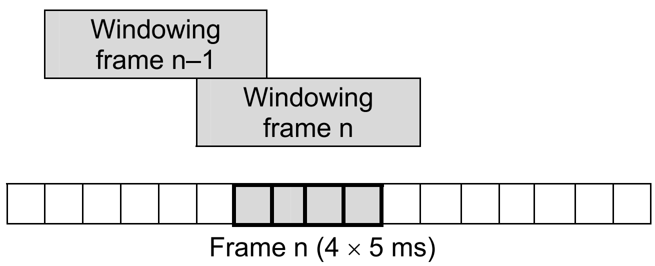

Figure 5.6-1: Relative positions and length of the LP analysis windows

The autocorrelation of windowed speech is converted to the LP coefficients using the Levinson-Durbin algorithm. Then the LP coefficients are transformed to the Immitance Spectral Pairs (ISP) domain for quantization and interpolation purposes. The interpolated quantized and unquantized filters are converted back to the LP filter coefficients (to construct the synthesis and weighting filters for each subframe).

5.6.1 Windowing and Autocorrelation Computation

The LP analysis is performed once per frame using an asymmetric window. The window is centered at the fourth subframe and it consists of two parts: the first part is a half of a Hamming window and the second part is a quarter of a cosine cycle. The window is given by

$$\begin{aligned} w(n) &= 0.54 - 0.46\cos\left(\frac{2\pi n}{2L_1 - 1}\right), & n &= 0,\ldots, L_1 - 1 \\ &= \cos\left(\frac{2\pi(n - L_1)}{4L_2 - 1}\right), & n &= L_1,.., L_1 + L_2 - 1 \end{aligned} \qquad (5.6.1\text{-}1)$$

where the values L_1=256 and L_2=128 are used.

The autocorrelations of the windowed speech $s''(n)= s'(n-64)\, w(n)$, n = 0,…,383 are computed by

$$r_c(k) = \sum_{n=k}^{383} s''(n)\, s''(n-k), \qquad k = 0,\dots,16 \tag{5.6.1-2}$$

and a 60 Hz bandwidth expansion is used by lag windowing the autocorrelations using the window

$$w_{lag}(i) = \exp\left[-\frac{1}{2}\left(\frac{2\pi f_0 i}{f_s}\right)^2\right], \qquad i = 1,\dots,16 \tag{5.6.1-3}$$

where f_0 = 60 Hz is the bandwidth expansion, f_s = 12800 Hz is the sampling frequency, and $r_c(0) \geq 100$. Furthermore, for wideband input signals, $r_c(0)$ is multiplied by the white noise correction factor 1.0001 which is equivalent to adding a noise floor at –40 dB. In case of narrowband inputs, $r_c(0)$ is multiplied by a stronger factor of 1.0031 to ease the LP coefficients estimation on a spectrum with a sharp cut-off frequency at the ends of the narrowband spectrum.

5.6.2 Levinson-Durbin Algorithm

The modified autocorrelations $r'(0) = 1.0001 r_c(0)$ or $r'(0) = 1.0031 r_c(0)$ and $r'(k) = r_c(k) w_{lag}(k), k = 1,\dots,16,$ are used to obtain the LP filter coefficients a_k, k = 1,…,16 by solving the set of equations.

$$\sum_{k=1}^{16} a_k r'(|i-k|) = -r'(i), \qquad i = 1,\dots,16 \tag{5.6.2-1}$$

The set of equations in (5.6.2-1) is solved using the Levinson-Durbin algorithm [21]. This algorithm uses the following recursion:

$$\begin{aligned}
& E(0) = r'(0) \\
& \text{For } i = 1 \text{ to } 16 \text{ do} \\
& \quad k_i = -\left[r'(i) + \sum\nolimits_{j=1}^{i-1} a_j^{i-1} r'(i-j)\right] / E(i-1) \\
& \quad a_i^{(i)} = k_i \\
& \quad \text{For } j = 1 \text{ to } i-1 \text{ do} \\
& \qquad a_j^{(i)} = a_j^{(i-1)} + k_i a_{i-j}^{(i-1)} \\
& \quad E(i) = (1 - k_i^2) E(i-1)
\end{aligned} \tag{5.6.2-2}$$

The final solution is given as $a_j = a_j^{(16)},\ j = 1,\dots,16$. The LP filter coefficients are converted to the ISP representation [22] for quantization and interpolation purposes. The conversions to the ISP domain and back to the LP filter domain are described in following sections.

5.6.3 LP to ISP conversion

The LP filter coefficients a_k, k = 1,…,16 are converted to the ISP representation for quantization and interpolation purposes. For a 16th order LP filter, the ISPs are defined as the roots of the sum and difference polynomials

$$f_1'(z) = A(z) + z^{-16}A(z^{-1}) \tag{5.6.3-1}$$

and

$$f_2'(z) = A(z) - z^{-16}A(z^{-1}) \tag{5.6.3-2}$$

respectively. (The polynomials $f_1'(z)$ and $f_2'(z)$ are symmetric and asymmetric, respectively). It can be proven that all roots of these polynomials are on the unit circle and interlaced. Polynomial $f_2'(z)$ has two roots at $z = 1$ ($\omega = 0$) and $z = -1$ ($\omega = \pi$). To eliminate these two roots, we define the new polynomials

$$f_1(z) = f_1'(z) \tag{5.6.3-3}$$

and

$$f_2(z) = f_2'(z)/(1 - z^{-2}) \tag{5.6.3-4}$$

Polynomials $f_1(z)$ and $f_2(z)$ have 8 and 7 conjugate roots on the unit circle $\left(e^{\pm j\omega_i}\right)$, respectively. Therefore, the polynomials can be written as

$$F_1(z) = (1 + a_{16}) \prod_{i=0,2,\ldots,14} (1 - 2q_i z^{-1} + z^{-2}) \tag{5.6.3-5}$$

and

$$F_2(z) = (1 + a_{16}) \prod_{i=1,3,\ldots,13} (1 - 2q_i z^{-1} + z^{-2}) \tag{5.6.3-6}$$

where $q_i = \cos(\omega_i)$ with ω_i being the Immitance Spectral Frequencies (ISF) and a_{16} is the last predictor coefficient. The ISFs satisfy the ordering property $0 < \omega_0 < \omega_1 < \ldots < \omega_{14} < \pi$. We refer to q_i as the ISPs in the cosine domain and ω_i as the ISFs in the frequency domain.

Since both polynomials $f_1(z)$ and $f_2(z)$ are symmetric only the first 8 and 7 coefficients of each polynomial, respectively, and the last predictor coefficient need to be computed. The coefficients of these polynomials are found by the following recursive relations

$f_1(8) = 2a_8$

for i = 0 to 7

$$\begin{aligned} f_1(i) &= a_i + a_{m-i} \\ f_2(i) &= a_i - a_{m-i} + f_2(i-2) \end{aligned} \tag{5.6.3-7}$$

where m = 16 is the predictor order, and $f_2(-2) = f_2(-1) = 0$.

The ISPs are found by evaluating the polynomials $F_1(z)$ and $F_2(z)$ at 100 points equally spaced between 0 and π and checking for sign changes. A sign change signifies the existence of a root and the sign change interval is then divided 4 times to more precisely track the root. The Chebyshev polynomials are used to evaluate $F_1(z)$ and $F_2(z)$ [23]. In this method the roots are found directly in the cosine domain $\{q_i\}$. The polynomials $F_1(z)$ and $F_2(z)$ evaluated at $z = e^{j\omega}$ can be written as

$$F_1(\omega) = 2e^{-j8\omega}C_1(x) \text{ and } F_2(\omega) = 2e^{-j7\omega}C_2(x) \tag{5.6.3-8}$$

with

$$C_1(x) = \sum_{i=0}^{7} f_1(i)T_{8-i}(x) + f_1(8)/2, \text{ and } C_2(x) = \sum_{i=0}^{6} f_2(i)T_{8-i}(x) + f_2(7)/2 \tag{5.6.3-9}$$

where $T_m = \cos(m\omega)$ is the mth order Chebyshev polynomial, $f(i)$ are the coefficients of either $F_1(z)$ or $F_2(z)$, computed using the equations in (5.6.3-7). The polynomial $C(x)$ is evaluated at a certain value of $x = \cos(\omega)$ using the recursive expression

$$\begin{aligned}
&\text{for } k = n_f - 1 \text{ down to } 1\\
&\quad b_k = 2xb_{k+1} - b_{k+2} + f(n_f - k)\\
&\text{end}\\
&C(x) = xb_1 - b_2 + f(n_f)/2
\end{aligned} \tag{5.6.3-10}$$

where $n_f = 8$ in case of $C_1(x)$ and $n_f = 7$ in case of $C_2(x)$, with initial values $b_{nf} = f(0)$ and $b_{nf+1} = 0$. The detail of the Chebyshev polynomial evaluation method can be found in [23].

5.6.4 ISP to LP Conversion

Once the ISPs are quantized and interpolated, the quantized and unquantized ISPs are converted back to the LP coefficient domain $\{a_k\}$. The conversion to the LP domain is done as follows. The coefficients of $F_1(z)$ and $F_2(z)$ are found by expanding Equations (5.6.3-5) and (5.6.3-6) knowing the quantized and interpolated ISPs q_i ,$i = 0,\ldots,m - 1$, where $m = 16$. The following recursive relation is used to compute $f_1(z)$

$$\begin{aligned}
&\text{for } i = 2 \text{ to } m/2\\
&\quad f_1(i) = -2q_{2i-2} f_1(i-1) + 2 f_1(i-2)\\
&\quad \text{for } j = i-1 \text{ down to } 2\\
&\quad\quad f_1(j) = f_1(j) - 2q_{2i-2} f_1(j-1) + f_1(j-2)\\
&\quad \text{end}\\
&\quad f_1(1) = f_1(1) - 2q_{2i-2}\\
&\text{end}
\end{aligned} \tag{5.6.4-1}$$

with initial values $f_1(0) = 1$ and $f_1(1) = -2q_0$. The coefficients $f_2(i)$ are computed similarly by replacing q_{2i-2} by q_{2i-1} and $m/2$ by $m/2-1$, and with initial conditions $f_2(0) = 1$ and $f_2(1) = -2q_1$.

Once the coefficients $f_1(z)$ and $f_2(z)$ are found, $F_2(z)$ is multiplied by $1-z^{-2}$, to obtain $F_2'(z)$; that is

$$\begin{aligned}
f_2'(i) &= f_2(i) - f_2(i-2), && i = 2,\ldots,m/2-1\\
f_1'(i) &= f_1(i) && i = 0,\ldots,m/2
\end{aligned} \tag{5.6.4-2}$$

Then $F_1'(z)$ and $F_2'(z)$ are multiplied by $1+q_{m-1}$ and $1-q_{m-1}$, respectively. That is

$$\begin{aligned}
f_2'(i) &= (1-q_{m-1}) f_2'(i), && i = 0,\ldots,m/2-1\\
f_1'(i) &= (1+q_{m-1}) f_1'(i) && i = 0,\ldots,m/2
\end{aligned} \tag{5.6.4-3}$$

Finally the LP coefficients are found by

$$
\begin{aligned}
a_i &= 0.5 f_1'(i) + 0.5 f_2'(i), && i = 1,\ldots,m/2-1 \\
& \quad 0.5 f_1'(i) - 0.5 f_2'(i), && i = m/2+1,\ldots m-1 \\
& \quad 0.5 f_1'(m/2), && i = m/2 \\
& \quad q_{m-1}, && i = m
\end{aligned}
\qquad (5.6.4\text{-}4)
$$

This is directly derived from the equation $A(z) = (F_1'(z) + F_2'(z))/2$, and considering the fact that $F_1'(z)$ and $F_2'(z)$ are symmetric and asymmetric polynomials, respectively.

5.6.5 Interpolation of ISPs

The set of LP parameters is used for the fourth subframe whereas the first, second, and third subframes use a linear interpolation of the parameters in the adjacent frames. The interpolation is performed on the ISPs in the **q** domain. Let $\mathbf{q}_4^{(n)}$ be the ISP vector at the 4th subframe of the frame, and $\mathbf{q}_4^{(n-1)}$ the ISP vector at the 4th subframe of the past frame *n*–1. The interpolated ISP vectors at the 1st, 2nd, and 3rd subframes are given by

$$
\begin{aligned}
\mathbf{q}_1^{(n)} &= 0.55\mathbf{q}_4^{(n-1)} + 0.45\mathbf{q}_4^{(n)} \\
\mathbf{q}_2^{(n)} &= 0.2\mathbf{q}_4^{(n-1)} + 0.8\mathbf{q}_4^{(n)} \\
\mathbf{q}_3^{(n)} &= 0.04\mathbf{q}_4^{(n-1)} + 0.96\mathbf{q}_4^{(n)}
\end{aligned}
\qquad (5.6.5\text{-}1)
$$

The same formula is used for interpolation of both quantized and unquantized ISPs. The interpolated ISP vectors are used to compute a different LP filter at each subframe (both quantized and unquantized) using the ISP to LP conversion method described in 5.6.4.

5.7 Perceptual Weighting

Routine Name: `find_wsp`
Inputs:

- *s* (*n*): The denoised input speech
- a_i : The LP filter coefficients

Outputs:

- s_w *(n)*: The perceptually weighted speech signal

Initialization:

- The memory of the filter is set to all zeros at initialization.

The encoding parameters such as adaptive codebook delay and gain, fixed-codebook index and gain are searched by minimizing the error between the input signal and synthesized signal in a perceptually weighted domain. Perceptual weighting is performed by filtering the signal through a perceptual weighting filter derived from the LP synthesis filter coefficients. The perceptual weighted signal is also used in open-loop pitch analysis and signal modification modules. The traditional perceptual weighting filter $W(z) = A(z/\gamma_1)/A(z/\gamma_2)$ has inherent limitations in modeling the formant structure and the required spectral tilt concurrently.

The spectral tilt is more pronounced in wideband signals due to the wide dynamic range between low and high frequencies. A solution to this problem is to introduce a pre-emphasis filter at the input for filtering the wideband signal to produce a pre-emphasized signal with enhanced high frequency content, calculate the LP synthesis filter coefficients from the pre-emphasized signal s(*n*), and produce the perceptually weighted speech signal by filtering the pre-emphasized signal through a perceptual weighting filter having a transfer function derived from the LP filter coefficients and with a denominator having fixed coefficients similar to the pre-emphasis filter so that the weighting of the wideband signal in the formant regions is decoupled from the spectral tilt of the wideband signal as will be shown below.

A weighting filter of the form

$$W(z) = A(z/\gamma_1)H_{\text{de-emph}}(z) = A(z/\gamma_1)/(1-\beta_1 z^{-1}) \tag{5.7-1}$$

is used, where $H_{\text{de-emph}} = \dfrac{1}{1-\beta_1 z^{-1}}$ and β_1 is fixed and equal to 0.68.

Because $A(z)$ is computed based on the pre-emphasized speech signal $s(n)$, the tilt of the filter $1/A(z/\gamma_1)$ is less pronounced compared to the case when $A(z)$ is computed based on the original speech (as the pre-emphasized signal itself exhibit less spectral tilt than the original wideband signal). Since de-emphasis is performed at the decoder end, it can be shown that the quantization error spectrum is shaped by a filter having a transfer function $W^{-1}(z)H_{\text{de-emph}}(z)=1/A(z/\gamma_1)$. Thus, the spectrum of the quantization error is shaped by a filter whose transfer function is $1/A(z/\gamma_1)$, with $A(z)$ computed based on the pre-emphasized speech signal. Figure 5.7-1 compares the noise shaping of the traditional and new weighting filters in case of an unvoiced speech segment. In the traditional case, the filter is given by $W'(z)=A'(z/0.9)/A'(z/0.6)$ where $A'(z)$ is computed using the original signal without pre-emphasis. The proposed filter is given by $W(z)=A(z/0.9)/(1-0.68z^{-1})$ where $A(z)$ is computed using the original signal after pre-emphasis with $1-0.68z^{-1}$. In the traditional case, the spectrum of the coding noise is shaped using the filter $1/W'(z)=A'(z/0.6)/A'(z/0.9)$ while in this case, it is shaped using the filter $1/A(z/0.9)$. Also in this case, the weighting filter shapes the coding noise in a way that it follows better the original speech spectrum. This is shown in Figure 5.7-1 where the traditional filter fails to shape the noise properly at low frequencies (about 15 dB difference between the two filters at low frequencies). This better noise shaping reflects the decoupling of the weighting from the spectral tilt.

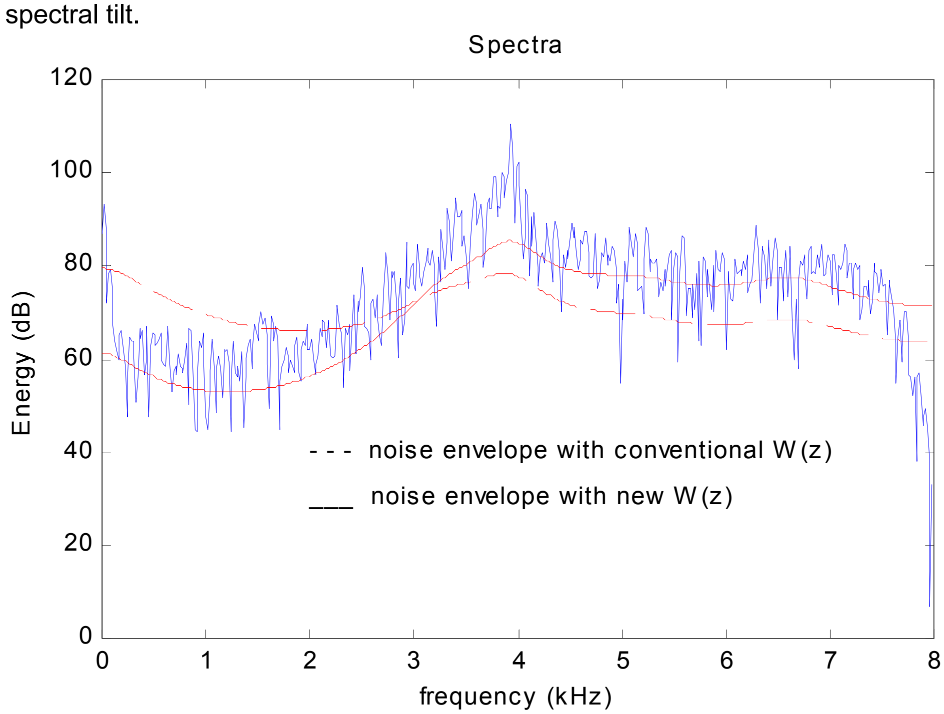

Figure 5.7-1: Spectrum of an unvoiced signal along with the quantization noise envelope after shaping by the conventional weighting filter and the new weighting filter.

The perceptual weighting is performed for the 20 ms frame while updating the LP filter coefficients on a 5-ms subframe basis using the interpolated filter parameters. In a subframe of size L, the weighted speech is given by

$$s_w(n) = s(n) + \sum_{i=1}^{16} a_i \gamma_1^i s(n-i) + \beta_1 s_w(n-1), \qquad n = 0,...,L-1 \tag{5.7-2}$$

Furthermore, for the open-loop pitch analysis, the computation is extended for a period of 10 ms using the lookahead from the future frame. This is done using the filter coefficients of the 4th subframe in the present frame. Note that this extended weighted signal is used only in the analysis of the present frame.

5.8 Open-loop Pitch Analysis and Pitch Tracking

<u>Routine Name</u>: `pitch_ol`

Inputs:

- $s_w(n)$: The weighted speech signal
- r_e: Noise correction factor
- E_{rel}: Relative frame energy

Outputs:

- d_0, d_1, and d_2 : The pitch lags in each half-frame
- $C_{norm}(d)$: The normalized correlation at pitch lags d_0, d_1, and d_2.

Initialization:

- The buffers and internal memories are set to zero at initialization.

Open-loop pitch analysis calculates three estimates of the pitch lag for each frame. This is done in order to smooth the pitch evolution contour and to simplify the pitch analysis by confining the closed-loop pitch search to a small number of lags around the open-loop estimated lags.

Open-loop pitch estimation is based on the weighted speech signal $s_w(n)$ computed as in Equation (5.7-2). The open-loop pitch analysis is performed on the weighted signal decimated by two. The decimated signal is obtained by filtering $s_w(n)$ through a 5th order FIR filter with coefficients {0.13, 0.23, 0.28, 0.23, 0.13} and then down-sampling the output by 2 to obtain the decimated weighted signal $s_{wd}(n)$.

Open-loop pitch analysis is performed 3 times per frame every 10 ms to find three estimates of the pitch lag: two in the present frame and one for the lookahead.

5.8.1 Correlation Function Computation

The pitch delay range (decimated by 2) is divided into the following four segments: [10,16], [17,31], [32, 61], and [62,115]. The first segment [10,16] is, however, used only under special circumstances to avoid quality degradation for pitch lags below the lowest pitch quantization limit. The autocorrelation function is first computed for each pitch lag value by

$$C(d) = \sum_{n=0}^{L_{\sec}} s_{wd}(n) s_{wd}(n-d), \tag{5.8.1-1}$$

where the summation limit depends on the delay section according to

$$\begin{array}{lll} L_{\sec} = 40 & \text{for} & d = 10,...,16 \\ L_{\sec} = 40 & \text{for} & d = 17,...,31 \\ L_{\sec} = 62 & \text{for} & d = 32,...,61 \\ L_{\sec} = 115 & \text{for} & d = 62,...,115 \end{array} \quad (5.8.1\text{-}2)$$

This will ensure that for a given delay value at least one pitch cycle is included in the correlation computation.

5.8.2 Correlation Reinforcement with Past Pitch Values

The autocorrelation function is then weighted to emphasize the function for delays in the neighborhood of pitch lag parameters determined from the previous frame, and the delay is identified based on the maximum of the weighted correlation function.

The weighting is given by a triangular window of size 27 and it is centered around the extrapolated pitch lags in the current frame based on lags determined in the previous frame. The extrapolated pitch lags are pitch lags estimated for the current frame in the neighborhood of those determined for the previous frame as will be shown below. The pitch neighborhood weighting function is given by

$$w_{pn}(13+i) = w_{pn}(13-i) = 1 + \alpha_{pn}(1 - i/14), \qquad i = 0,...,13. \quad (5.8.2\text{-}1)$$

where α_{pn} is a scaling factor based on the voicing measure from the previous frame (the normalized pitch correlation) and the pitch stability in the previous frame. During voiced segments with smooth pitch evolution, the scaling factor is updated from previous frame by adding a value of $0.16\overline{R}_{xy}$, and it is upper-limited to 0.7. $\overline{R}_{xy}$ is the average of the normalized correlation in the two half frames of the previous frame. The scaling factor α_{pn} is reset to zero (no weighting) if $\overline{R}_{xy}$ is less than 0.4 or if the pitch lag evolution in the previous frame is unstable or if the difference between the energy of the previous frame and the long-term average energy of active speech is more than a certain threshold. The pitch instability is determined by testing the pitch coherence between consecutive half frames. The pitch values of two consecutive half frames are considered coherent if the following condition is satisfied:

(max_value < 1.4 min_value) AND ((max_value – min_value) < 14)

where max_value and min_value denote the maximum and minimum of the two pitch values, respectively. The pitch evolution in a certain frame is considered as stable if pitch coherence is satisfied for both the first half frame and the last half frame of previous frame as well as the two halves of the present frame.

The extrapolated pitch lag in the first half frame, $\tilde{d}_0$, is computed as the pitch lag from the second half frame of the previous frame plus a pitch evolution factor f_{evol} computed from the previous frame (as described in Section 5.8.6). The extrapolated pitch lag in the second half frame, $\tilde{d}_1$, is computed as the pitch lag from the second half frame of the previous frame plus twice the pitch evolution factor. That is

$$\tilde{d}_0 = d_{-1} + f_{evol} \quad \text{and} \quad \tilde{d}_1 = d_{-1} + 2 f_{evol} \quad (5.8.2\text{-}2)$$

where d_{-1} is the pitch lag in the second half-frame of previous frame.

The pitch evolution factor is obtained by averaging the pitch differences of consecutive half frames that are determined as coherent (according to the coherence rule described above). The autocorrelation function weighted around the extrapolated pitch lag $\tilde{d}$ is given by

$$C^{w}(\tilde{d}+i) = C(\tilde{d}+i)w_{pn}(13+i), \qquad i=-13,...,13 \tag{5.8.2-3}$$

5.8.3 Normalized Correlation Computation

After weighting the correlation function with the triangular window of Equation (5.8.2-1) centered at the extrapolated pitch lag, the maxima of the weighted correlation function in each of the N_{sec} sections described above, $C^{w}(d_{\max}^{(i)})$, i=0,1, 2 and 3, are determined. N_{sec} equals 3 or 4, depending whether section 0 is used. Section 0 is used only during stable high-pitched speech segments, i.e. if the open-loop pitch period of the 2nd half frame of the previous frame is lower or equal to 24 and the scaling factor α_{pn} is higher or equal to 0.1. The correlations at N_{sec} lag positions (for the N_{sec} sections) are normalized according to

$$C_{norm}(d_{\max}) = \frac{C(d_{\max})}{\sqrt{\sum_{n=0}^{L_{\sec}} s_{wd}^{2}(n) \sum_{n=0}^{L_{\sec}} s_{wd}^{2}(n-d_{\max})}}, \tag{5.8.3-1}$$

where the summation limit depends on the pitch delay section as defined in Equation (5.8.1-2). At this point, N_{sec} candidate pitch lags $d_{\max}^{(k)}(i)$ have been determined (one per section) for each half frame and look-ahead with corresponding values of normalized correlations (both weighted and raw), where *i* is the half-frame index and *k* is the section index. All remaining processing is performed using only these selected values, greatly reducing the overall complexity.

Note that the last section (long pitch periods) is not searched for the look-ahead. Instead, the normalized correlation values and the corresponding pitch are obtained from the last section search of the 2nd half frame. The reason is that the summation limit in the last section is much larger than the available lookahead and a limitation of the computational complexity.

5.8.4 Correlation Reinforcement with Pitch Lag Multiples

In order to avoid selecting pitch multiples, the weighted normalized correlation in a lower pitch delay section is further emphasized if one of its multiples is in the neighborhood of the pitch lag at the maximum weighted correlation in a higher section. That is,

$$\text{If } \left(\left|k \times d_{\max}^{(2)} - d_{\max}^{(3)}\right| \le k\right) \text{ then } C_{norm}^{w}(d_{\max}^{(2)}) = \alpha_{mult} C_{norm}^{w}(d_{\max}^{(2)}),\ \alpha_{mult} = (\alpha_{mult})^{2}$$

$$\text{If } \left(\left|k \times d_{\max}^{(1)} - d_{\max}^{(2)}\right| \le k\right) \text{ then } C_{norm}^{w}(d_{\max}^{(1)}) = \alpha_{mult} C_{norm}^{w}(d_{\max}^{(1)}),$$

where $\alpha_{mult}=1.17$. In this way, if a pitch period multiple is found in a higher section, the maximum weighted correlation in the lower section is emphasized by a factor 1.17. If however the maximum correlation in sections 3 is at pitch period multiple of the maximum correlation of the section 2 and the maximum correlation in sections 2 is at pitch period multiple of the maximum correlation of the section 1, the maximum weighted correlation in section 1 is emphasized twice.

It can be seen that the “neighborhood” is larger with the number of multiples *k* to take into account an increasing uncertainty on the pitch length (the pitch length is estimated roughly with integer precision at 6400 Hz sampling frequency). The uncertainty is even higher for the section 3 of the lookahead, when the maximum correlation value and the corresponding pitch period were simply copied from the corresponding section of the 2nd half frame. For this reason, the condition in the first relation above is modified for lookahead as follows

$$\text{If } \left(\left| k \times d_{\max}^{(1)} - d_{\max}^{(2)} \right| \le 2(k-1) \right)$$

Note that section 0 is not considered here, i.e. the maximum normalized correlation in section 0 is never emphasized.

5.8.5 Initial Pitch Lag Determination and Reinforcement Based on Pitch Coherence with other Half-frames

For each half frame and lookahead, an initial pitch lag $d_{init}(i)$, *i* being half-frame index, and corresponding raw (unweighted) normalized correlation are determined by searching the maximum of N_{sec} emphasized normalized correlations (corresponding to N_{sec} sections).

In the previously described pitch search and tracking, the correlations were weighted at the neighborhood of a pitch lag extrapolated from previous frame pitch lags. Now, to further track the right pitch value, another level of weighting is performed of the correlations in each section and each half-frame based on the pitch coherence of initially determined pitch lags $d_{init}(i)$ with pitch lags $d_{\max}^{(k)}(j \neq i)$ of each section in the other half-frames. That is, if the initial pitch lag in a half-frame *i* is coherent with pitch lag of section *k* in half-frame *j*, then the corresponding weighted normalized correlation of section *k* in half-frame *j* is further emphasized by weighting it by the value $1+\alpha(1-\delta_{pit}/14)$ where δ_{pit} is the absolute difference between $d_{init}(i)$ and $d_{\max}^{(k)}(j \neq i)$, and $\alpha = 0.4(C_{norm}(d_{init}^{(i)}) + 0.5r_e)$ where α is upper-bounded to 0.4, $C_{norm}(d)$ is the normalized correlation as defined in Equation (5.8.3-1), but without weighting and r_e is a correction added to the normalized correlation in order to compensate for the decrease of normalized correlation in the presence of background noise (defined in Equation (5.4.4-1)). This procedure will further help in avoiding selecting pitch multiples and insure pitch continuity in adjacent half-frames.

5.8.6 Pitch Lag Determination and Parameter Update

Finally, the pitch lags in each half-frame, d_0, d_1, and d_2, are determined. Again, the further emphasized normalized correlation in the last section of look-ahead is copied from the last section of the 2nd half frame before computing the final pitch lags. They are determined by searching the maximum of the emphasized normalized correlations corresponding to each of the N_{sec} sections. After determining the pitch lags, the parameters needed for the next frame pitch search are updated. The average normalized correlation $\overline{R}_{xy}$ is updated by:

$$\overline{R}_{xy} = 0.5(C_{norm}(d_0) + C_{norm}(d_1)) + 0.5r_e \quad \text{constrained by } \overline{R}_{xy} \le 1, \tag{5.8.6-1}$$

Finally, the pitch evolution factor f_{evol} to be used in computing the extrapolated pitch lags in the next frame is updated. The pitch evolution factor is given by averaging the pitch differences of consecutive half frames that are determined as coherent. If d_{-1} is the pitch lag in the second half of the previous frame then pitch evolution is given by

$$\delta_{pitch} = 0$$

$cnt = 0$

For i=0 to 2 do

if d_i and d_{i-1} are coherent

$$\delta_{pitch} = \delta_{pitch} + d_i - d_{i-1}$$

$$cnt = cnt + 1$$

if $(cnt > 0)$ $f_{evol} = \delta_{pitch} / cnt$ else $f_{evol} = 0$

Since the search is performed on the decimated weighted signal, the determined pitch lags d_0, d_1, and d_2 are multiplied by 2 to obtain the open loop pitch lags for the 3 half-frames.

If a selected pitch lag value is lower or equal to the minimum pitch quantization level (34 samples at 12800 Hz sampling frequency), the pitch value is multiplied by two. The reason is that the calculated pitch value can saturate to this minimum pitch lag when the real pitch period is below that value. This may occur in case of high-pitched female or child speakers. Pitch periods under the lower limit of 34, but very close to it, would saturate into 34 causing constant pitch period to be generated in the decoder and consequently quality degradation.

5.9 Noise Energy Estimate Update and Voiced Critical Band Determination

Routine Name: `noise_est`

Inputs:

- $E_{CB}(i)$: Average energy in ith critical band
- $N_{CB}(i)$: Noise estimate in ith critical band
- r_e: Noise correction factor
- d_0, d_1, and d_2 : The pitch lags in each half-frame
- $C_{norm}(d)$: The normalized correlation at pitch lags d_0, d_1, and d_2.
- E_t: Total frame energy
- $E(i)$: LP residual energies

Outputs:

- $N_{CB}(i)$: Noise estimate in ith critical band
- K_{voic}: Voiced critical bands

Initialization:

- $N_{CB}(i)$ and $E_{CB,LT}(i)$ are initialized to 0.03 (unless otherwise described for $N_{CB}(i)$ in Section 5.4. Previous frame pitch lag d_{-1} is initialized to zero. The noise update decision hangover is initialized to 6.

This module updates the noise energy estimates per critical band for noise suppression. The update is performed during inactive speech intervals. However, the VAD decision obtained in Section 5.4, which is based on the SNR per critical band, is not used for determining whether the noise energy estimates are updated. Another decision is performed based on other parameters independent of the SNR per critical band. The parameters used for the noise update decision are: pitch stability, signal non-stationarity, voicing, and ratio between 2nd order and 16th order LP residual error energies. These parameters have generally low sensitivity to the noise level variations.

The reason for not using the encoder VAD decision for noise update is to make the noise estimation robust to rapidly changing noise levels. If the encoder VAD decision were used for the noise update, a sudden increase in noise level would cause an increase of SNR even for inactive speech frames, preventing the noise estimator to update, which in turn would maintain the SNR high in following frames. Consequently, the noise update would be blocked and some other logic would be needed to resume the noise adaptation.

The pitch stability counter is computed as

$$pc = | d_0 - d_{-1} | + | d_1 - d_0 | + | d_2 - d_1 | \qquad (5.9\text{-}1)$$

where d_0, d_1, and d_2 ,are the open-loop pitch lags for the first half-frame, second half-frame, and the lookahead, respectively, and d_{-1} is the lag of the second half-frame of the pervious frame. Since for pitch lags larger than 122, the open-loop pitch search module sets $d_2 = d_1$, then for such lags the value of pc in equation (5.9-1) is multiplied by 3/2 to compensate for the missing third term in the equation. The pitch stability is true if the value of pc is less than 12. Further, for frames with low voicing, pc is set to 12 to indicate pitch instability. That is

$$\text{If } (C_{norm}(d_0) + C_{norm}(d_1) + C_{norm}(d_2))/3 + r_e < 0.7 \text{ then } pc = 12, \qquad (5.9\text{-}2)$$

where $C_{norm}(d)$ is the normalized raw correlation as defined in Equation (5.8.3-1) but without weighting and r_e is a correction added to the normalized correlation in order to compensate for the decrease of normalized correlation in the presence of background noise (defined in Equation (5.4.4-1)).

The signal non-stationarity estimation is performed based on the product of the ratios between the energy per critical band and the average long-term energy per critical band.

The average long-term energy per critical band is updated by

$$E_{CB,LT}(i) = \alpha_e E_{CB,LT}(i) + (1-\alpha_e)\overline{E}_{CB}(i), \qquad \text{For } i=b_{min} \text{ to } b_{max}, \qquad (5.9\text{-}3)$$

where b_{min}=0 and b_{max}=19 in case of wideband signals, and b_{min}=1 and b_{max}=16 in case of narrowband signals, and $\overline{E}_{CB}(i)$ is the frame energy per critical band defined in Equation (5.4.2-1). The update factor α_e is a linear function of the total frame energy, defined in Equation (5.2-7), and it is given as follows:

For wideband signals: $\alpha_e = 0.0245 E_{tot} - 0.235$ constrained by $0.5 \le \alpha_e \le 0.99$.

For narrowband signals: $\alpha_e = 0.00091 E_{tot} + 0.3185$ constrained by $0.5 \le \alpha_e \le 0.999$.

The frame non-stationarity is given by the product of the ratios between the frame energy and average long-term energy per critical band. That is

$$nonstat = \prod_{i=b_{min}}^{b_{max}} \frac{\max(\overline{E}_{CB}(i), E_{CB,LT}(i))}{\min(\overline{E}_{CB}(i), E_{CB,LT}(i))} \qquad (5.9\text{-}4)$$

The voicing factor for noise update is given by

$$voicing = (C_{norm}(d_0) + C_{norm}(d_1))/2 + r_e \; . \quad (5.9\text{-}5)$$

Finally, the ratio between the LP residual energy after 2nd order and 16th order analysis is given by

$$resid\_ratio = E(2)/E(16) \quad (5.9\text{-}6)$$

where E(2) and E(16) are the LP residual energies after 2nd order and 16th order analysis, and computed in the Levinson-Durbin recursion of Equation (5.6.2-2). This ratio reflects the fact that to represent a signal spectral envelope a higher order of LP is generally needed for speech signal than for noise. In other words, the difference between *E*(2) and *E*(16) is expected to be lower for noise than for active speech.

The update decision is determined based on a variable *noise_update*, which is initially set to 6, and it is decremented by 1 if an inactive frame is detected and incremented by 2 if an active frame is detected. Further, *noise_update* is bounded by 0 and 6. The noise energies are updated only when *noise_update*=0.

The value of the variable *noise_update* is updated in each frame as follows:

If (*nonstat* > th_{stat}) OR (*pc* < 12) OR (*voicing* > 0.85) OR (*resid_ratio* > th_{resid})
noise_update = *noise_update* + 2
Else
noise_update = *noise_update* – 1

where for wideband signals, th_{stat}=350000 and th_{resid}=1.9, and for narrowband signals, th_{stat}=500000 and th_{resid}=11.

In other words, frames are declared inactive for noise update when

(*nonstat* ≤ th_{stat}) AND (*pc* ≥ 12) AND (*voicing* ≤ 0.85) AND (*resid_ratio* ≤ th_{resid})

and a hangover of 6 frames is used before noise update takes place.

Thus, if *noise_update*=0 then

for i=0 to 19 $N_{CB}(i) = N_{tmp}(i)$

where $N_{tmp}(i)$ is the temporary updated noise energy already computed in Equation (5.4.2-4).

5.9.1 Update of Voicing Cutoff Frequency

The cut-off frequency below which a signal is considered voiced is updated. This frequency is used to determine the number of critical bands for which noise suppression is performed using per bin processing.

First, a voicing measure is computed as

$$v_g = 0.4C_{norm}(d_1) + 0.6C_{norm}(d_2) + r_e \quad (5.9.1\text{-}1)$$

and the voicing cut-off frequency is given by

$$f_c = 0.00017118\, e^{17.9772 v_g} \qquad \text{constrained by } 325 \le f_c \le 3700 \quad (5.9.1\text{-}2)$$

Then, the number of critical bands, K_{voic}, having an upper frequency not exceeding f_c is determined. The bounds of $325 \le f_c \le 3700$ are such that per bin processing is performed on a minimum of 3 bands and a maximum of 17 bands (refer to the critical bands upper limits in Section 5.2). Note that in the voicing measure calculation, more weight is given to the normalized correlation of the lookahead since the determined number of voiced bands will be used in the next frame.

Thus, in the following frame, for the first K_{voic} critical bands, the noise suppression will use per bin processing as described in Section 5.5.

Note that for frames with low voicing and for large pitch delays, only per critical band processing is used and thus K_{voic} is set to 0. The following condition is used:

If $(0.4C_{norm}(d_1) + 0.6C_{norm}(d_2) \le 0.72)$ OR $(d_1 > 116)$ OR $(d_2 > 116)$ then $K_{voic} = 0$.

5.10 Unvoiced Signal Classification: Selection of Unvoiced-HR and Unvoiced-QR

<u>Routine Name</u>: `rate_select`

Inputs:

- *s* (*n*): The denoised speech signal
- Mode of operation and HR-maximum signaling flag
- $C_{norm}(d)$: The normalized correlation at pitch lags d_0, d_1, and d_2.
- VAD_flag and local VAD flag
- r_e: Noise correction factor
- E_{rel}: Relative frame energy
- $\overline{N}_f$: Long-term average noise energy
- Previous frame encoding scheme

Outputs:

- $e_{tilt}(i)$: spectral tilt
- Encoding type

Initialization:

- e_{old}: The tilt in the second half of the previous frame is initialized to 10. Previous frame encoding type is initialized to Generic FR.

Figure 5.10-1 shows a simplified high-level description of the signal classification procedure. If voice activity is not detected, CNG-ER encoding type is utilized. If voice activity is detected, the voiced versus unvoiced classification is performed. If the frame is classified as unvoiced, it is encoded with either Unvoiced HR or Unvoiced QR encoding types. If the frame is not classified as unvoiced, then stable voiced classification is applied. If the frame is classified as stable voiced, it can be encoded using Voiced HR encoding type. Otherwise, the frame is likely to contain a non-stationary speech segment such as a voiced onset or rapidly evolving voiced speech signal. These frames typically require a general-purpose coding model at high bit rate for sustaining good speech quality. Thus in this case an appropriate FR encoding type is mainly used. Frames with very low energy and not detected as non-speech, unvoiced or stable voiced can be encoded using Generic HR coding in order to reduce the average data rate.

This is a simplified description of the rate determination procedure. The actual choice of encoding type in a certain frame is based both on the frame classification and the required mode of operation. In VMR-WB mode 0 for instance, HR Voiced encoding type is not used and the Unvoiced QR encoding type is used only in VMR-WB mode 2. Thus, the classification thresholds depend on the VMR-WB mode of operation. Furthermore, the rate determination procedure has to comply with maximum and minimum rate constraints.

The VAD has been described in Section 5.3. In this section, unvoiced signal classification will be described. The stable voiced signal classification will be described as part of the signal modification procedure (see Section 5.11.4). Finally, the last stage of classification to determine if the frame can be encoded using Generic HR encoding type is described in Section 5.12.

The unvoiced parts of the signal are characterized by missing periodic component and can be further divided into unstable frames, where the energy and the spectrum changes rapidly, and stable frames where these characteristics remain relatively stable. The classification of unvoiced frames exploits the following parameters:

- A voicing measure, computed as an averaged normalized correlation, $\overline{R}_{xy3}$,
- Spectral tilt measures, $e_{tilt}(0)$ and $e_{tilt}(1)$, for both spectral analysis per frame,
- A signal energy ratio (dE) used to assess the frame energy variation within the frame and thus the frame stability, and
- Relative frame energy

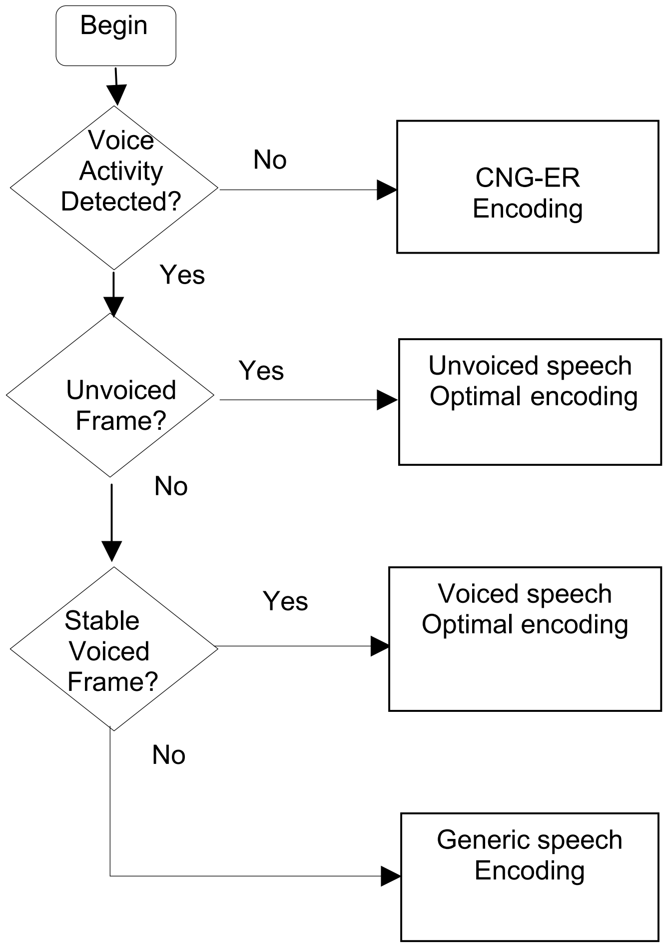

Figure 5.10-1: Functional block diagram of the high-level signal classification logic.

5.10.1 Voicing Measure

The normalized correlation, used to determine the voicing measure, is computed as part of the open-loop search module described in Section 5.8. The normalized correlation is computed similar to Equation (5.8.6-1) with also considering the lookahead half-frame. That is

$$\overline{R}_{xy3} = \frac{1}{3}(C_{norm}(d_0) + C_{norm}(d_1) + C_{norm}(d_2)) + r_e \qquad (5.10.1\text{-}1)$$

where $C_{norm}(d)$ is computed as in Equation (5.8.3-1) and r_e as in Equation (5.4.4-1).

5.10.2 Spectral Tilt

The spectral tilt parameter contains the information about the frequency distribution of energy. The spectral tilt is estimated in the frequency domain as a ratio between the energy concentrated in low frequencies and the energy concentrated in high frequencies.

The energy in high frequencies is computed as the average of the energies of the last two critical bands

$$\overline{E}_h = 0.5[E_{CB}(b_{\max} - 1) + E_{CB}(b_{\max})] \qquad (5.10.2\text{-}1)$$

where $E_{CB}(i)$ are the critical band energy computed in Equation (5.2-4) and b_{max}=19 for WB inputs and b_{max}=16 for NB inputs.

The energy in low frequencies is computed as the average of the energies in the first 9 critical bands. The middle critical bands have been excluded from the computation to improve the discrimination between frames with high-energy concentration in low frequencies (generally voiced) and with high-energy concentration in high frequencies (generally unvoiced). In between, the energy content is not informative for any of the classes and increases the decision uncertainty.

The energy in low frequencies is computed differently for long pitch periods and short pitch periods. For voiced female speech segments, the harmonic structure of the spectrum is exploited to increase the voiced-unvoiced discrimination. Thus for short pitch periods, E_l is computed bin-wise and only frequency bins sufficiently close to the speech harmonics are taken into account in the summation. That is

$$\overline{E}_l = \frac{1}{cnt}\sum_{k=K_{\min}}^{24} E_{BIN}(k)w_h(k) \qquad (5.10.2\text{-}2)$$

where K_{min} is the first bin (K_{min}=0 for WB inputs and K_{min}=2 for NB inputs) and $E_{BIN}(k)$ are the bin energies, as defined in Equation (5.2-5), in the first 25 frequency bins (the DC component is not considered). Note that these 25 bins correspond to the first 10 critical bands and that the first 2 bins not included in the case of NB input constitute the 1st critical band. In the summation above, only the terms related to the bins close to the pitch harmonics are considered, so $w_h(k)$ is set to 1, if the distance between the nearest harmonics is not larger than a certain frequency threshold (50 Hz) and is set to 0 otherwise. The counter *cnt* is the number of the non-zero terms in the summation. Only bins closer than 50 Hz to the nearest harmonics are taken into account. Hence, if the structure is harmonic in low frequencies, only high-energy term will be included in the sum. On the other hand, if the structure is not harmonic, the selection of the terms will be random and the sum will be smaller. Thus even unvoiced sounds with high energy content in low frequencies can be detected. This processing cannot be done for longer pitch periods, as the frequency resolution is not sufficient. For pitch values larger than 128 or for a priori unvoiced sounds the low frequency energy is computed per critical band as

$$\overline{E}_l = \frac{1}{10}\sum_{k=0}^{9} E_{CB}(k) \quad \text{or} \quad \overline{E}_l = \frac{1}{9}\sum_{k=1}^{9} E_{CB}(k) \qquad (5.10.2\text{-}3)$$

for WB and NB inputs, respectively. A priori unvoiced sounds are determined when $\frac{1}{2}(C_{norm}(d_1) + C_{norm}(d_3)) + r_e < 0.6$.

The resulting low and high frequency energies are obtained by subtracting estimated noise energy from the values $\overline{E}_l$ and $\overline{E}_h$ calculated above. That is

$$E_h = \overline{E}_h - N_h \qquad (5.10.2\text{-}4)$$

$$E_l = \overline{E}_l - N_l \qquad (5.10.2\text{-}5)$$

where N_h and N_l are the averaged noise energies in the last 2 critical bands and first 10 critical bands (or 9 for NB inputs), respectively, computed similar to Equations (5.10.2-1) and (5.10.2-3). The estimated noise energies have been added to the tilt computation to account for the presence of background noise.

Finally, the spectral tilt is given by

$$e_{tilt}(i) = \frac{E_l}{E_h} \quad , \tag{5.10.2-6}$$

Note that the spectral tilt computation is performed twice per frame to obtain $e_{tilt}(0)$ and $e_{tilt}(1)$ corresponding to both spectral analysis per frame. The average spectral tilt used in unvoiced frame classification is given by

$$e_t = \frac{1}{3}\left(e_{old} + e_{tilt}(0) + e_{tilt}(1)\right) \tag{5.10.2-7}$$

where e_{old} is the tilt in the second half of the previous frame

5.10.3 Energy Variation

The energy variation *dE* is evaluated on the denoised speech signal *s* (*n*), where *n*=0 corresponds to the beginning of the current frame. The signal energy is evaluated twice per subframe, i.e. 8 times per frame, based on short-time segments of length 32 samples. Further, the short-term energies of the last 32 samples from the previous frame and the first 32 samples from next frame are also computed. The short-time maximum energies are computed as

$$E_{st}^{(1)}(j) = \max_{i=0}^{31}\left(s^2(i+32j)\right), \qquad j = -1,\ldots,8, \tag{5.10.3-1}$$

where j=-1 and j=8 correspond to the end of previous frame and the beginning of next frame. Another set of 9 maximum energies is computed by shifting the speech indices in Equation (5.10.3-1) by 16 samples. That is

$$E_{st}^{(2)}(j) = \max_{i=0}^{31}\left(s^2(i+32j-16)\right), \qquad j = 0,\ldots,8, \tag{5.10.3-2}$$

The maximum energy variation *dE* is computed as the maximum of the following:

$$E_{st}^{(1)}(0) / E_{st}^{(1)}(-1)$$

$$E_{st}^{(1)}(7) / E_{st}^{(1)}(8)$$

$$\frac{\max(E_{st}^{(1)}(j), E_{st}^{(1)}(j-1))}{\min(E_{st}^{(1)}(j), E_{st}^{(1)}(j-1))} \qquad \text{For j=1 to 7}$$

$$\frac{\max(E_{st}^{(2)}(j), E_{st}^{(2)}(j-1))}{\min(E_{st}^{(2)}(j), E_{st}^{(2)}(j-1))} \qquad \text{For j=1 to 8}$$

5.10.4 Relative Frame Energy E_{rel}

The relative frame energy is used to identify low energy frames where the unvoiced classification thresholds can be relaxed, allowing these frames to be classified as unvoiced more easily. The relative frame energy is also used later to identify low energy frames, which has not been classified as background noise frames or unvoiced frames. These frames can be encoded with a generic HR encoding type in order to reduce the average data rate (Section 5.12). The relative frame energy is computed in Equation (5.4.1-2).

5.10.5 Unvoiced Speech Classification

The classification of unvoiced speech frames is based on the parameters described above, namely: the voicing measure $\overline{R}_{xy3}$, the spectral tilt e_t, the energy variation within a frame dE, and the relative frame energy E_{rel}. The decision is made based on at least three of these parameters. The decision thresholds are set based on the operating mode. For operating modes with lower allowable data rates, the thresholds are set to favor more unvoiced classification (since a half-rate or a quarter-rate encoding type will be used to encode the frame). Unvoiced frames are usually encoded with Unvoiced HR encoder. However, in VMR-WB mode 2, Unvoiced QR is also used in order to further reduce the average data rate, if additional certain conditions are satisfied.

In VMR-WB mode 0, the frame is encoded as Unvoiced HR if the following condition is satisfied

$$(\overline{R}_{xy3} < th_1) \text{ AND } (e_t < th_2) \text{ AND } (dE < th_3)$$

where $th_1 = 0.5$, $th_2 = 1$, and th_3 = 0 if the long-term average noise energy $\overline{N}_f$ > 21 and th_3 = 4 otherwise. Furthermore, for WB inputs only, th_3 = 3.2 if $\overline{N}_f$ > 34.

In VAD, a decision hangover is used. Thus, after active speech periods, when the algorithm decides that the frame is an inactive speech frame, a local VAD is set to zero but the actual VAD_flag is set to zero only after a certain number of frames are elapsed (the hangover period). This avoids clipping of speech offsets. In both VMR-WB modes 1 and 2, if the local VAD is zero, the frame is classified as an Unvoiced frame.

In VMR-WB mode 1, the frame is encoded as Unvoiced HR if local VAD=0 OR if the following condition is satisfied

$$(\overline{R}_{xy3} < th_4) \text{ AND } (e_t < th_5) \text{ AND } ((dE < th_6) \text{ OR } (E_{rel} < th_7))$$

where $th_4 = 0.695$, $th_5 = 4$, $th_6 = 40$, and $th_7 = -14$. Note that dE is increased by 34 in case of NB inputs.

In VMR-WB mode 2, the frame is declared as an Unvoiced frame if local VAD=0 OR if the following condition is satisfied

$$(\overline{R}_{xy3} < th_8) \text{ AND } (e_t < th_9) \text{ AND } ((dE < th_{10}) \text{ OR } (E_{rel} < th_{11}))$$

where $th_8 = 0.695$, $th_9 = 4$, $th_{10} = 60$, and $th_{11} = -14$, $th_7 = -14$. Note that dE is increased by 10 in case of NB inputs and if $\overline{N}_f$ > 21.

In VMR-WB mode 2, unvoiced frames are usually encoded as Unvoiced HR. However, they can also be encoded with Unvoiced QR, if the following further conditions are also satisfied: If the last frame is either unvoiced or background noise frame, and if at the end of the frame the energy is concentrated in high frequencies and no potential voiced onset is detected in the lookahead then the frame is encoded as Unvoiced QR. The last two conditions are detected as:

$$(C_{norm}(d_2) < th_{12}) \text{ AND } (e_{tilt}(1) < th_{13}) \text{ where } th_{12} = 0.73,\ th_{13} = 3.$$

Note that $R_{xy3}(2)$ is the normalized correlation in the lookahead and $e_{tilt}(1)$ is the tilt in the second spectral analysis, which spans the end of the frame and the lookahead.

5.11 Signal Modification and HR Voiced Rate Selection

Routine Name: `sig_modification`
Inputs:

- $s(n)$: The denoised input speech
- $s_w(n)$: The weighted speech signal
- a_i : The LP filter coefficients
- Mode of operation and HR-maximum signaling flag
- E_{rel}: Relative frame energy
- $\overline{N}_f$: Long-term average noise energy
- d_i, $\overline{d}_{k-1}$: The open-loop pitch lags in current and previous frames and the signal modification pitch delay parameter at the frame end of previous frame boundary or the close-loop pitch of the last subframe.
- $\hat{s}_w(n)$: The weighted synthesized signal from the previous frame

Outputs:

- $s(n)$: The modified speech signal
- Encoding scheme
- $\overline{d}_k$: The signal modification pitch delay parameter at the frame end boundary

Initialization:

- The buffers and filter memories are reset at initialization. The previous frame pitch values are initialized to 50. The last pitch pulse position T_0 in the previous frame is initialized to –10 with respect to the beginning of the current frame.

The signal modification algorithm performs an inherent classification of voiced frames. It is used only if the current frame has not been classified so far, that is if it has not been classified as inactive speech frame or Unvoiced frame. The signal is actually modified only if the classification procedure selected the Voiced HR encoding type. This encoding type is generally used only in VMR-WB modes 1 and 2 and under the condition that the previous frame type is FR or Voiced HR. In VMR-WB mode 0, Voiced HR is not used in normal operation. However, in maximum half-rate operation, VMR-WB mode 0 can use the Voiced HR encoding type in case of stable voiced frames.

Signal modification is adjusting the speech signal LP residual to the determined delay contour $\tilde{d}(n)$. The delay contour $\tilde{d}(n)$ defines a long-term prediction delay for every sample of the frame. The delay contour is fully characterized over the frame $k \in (t_{k-1}, t_k]$ by a delay parameter $\overline{d}_k = \tilde{d}(t_k)$ and its previous value $\overline{d}_{k-1} = \tilde{d}(t_{k-1})$ that are equal to the value of the delay contour at frame boundaries. For a frame k with size L=256, the frame boundaries are $t_k = L$-1 and t_{k-1}=-1, corresponding to the last samples in the present frame and previous frame, respectively. The delay parameter is determined as a part of the signal modification procedure and encoded once per frame.

The signal modification is performed prior to the closed-loop pitch search of the adaptive codebook excitation signal that is prior to the ACELP subframe loop. The delay contour $\tilde{d}(n)$ defining a long-term prediction delay parameter for every sample of the frame is supplied to an adaptive codebook. The delay contour $\tilde{d}(n)$ is used to form the adaptive codebook excitation *v*(*n*) corresponding to the current subframe from the excitation *u*(*n*) using the delay contour $\tilde{d}(n)$ as $v(n) = u(n - \tilde{d}(n))$. Thus the delay contour maps the past sample of the excitation signal $u(n - \tilde{d}(n))$ to the present sample in the adaptive codebook excitation *v*(*n*).

The signal modification procedure produces a modified residual signal $\breve{r}(n)$ to be used for composing a modified target signal for the closed-loop search of the fixed-codebook excitation *c*(*n*). The modified residual signal $\breve{r}(n)$ is obtained by shifting the pitch cycle segments of the LP residual signal. The LP synthesis filtering of the modified residual signal with the filter 1/*A*(*z*) then yields the modified speech signal. The modified target signal of the fixed-codebook excitation search is formed in accordance with the ACELP operation, but with the original speech signal replaced by its modified version. After the adaptive codebook excitation *v*(*n*) and the modified target signal have been obtained for the current subframe, the encoding can further proceed using the methods described in the next sections.

When signal modification is enabled, the speech decoder recovers the delay contour $\tilde{d}(n)$ using the received delay parameter $\bar{d}_k$ and its previous received value $\bar{d}_{k-1}$ as in the encoder. This delay contour $\tilde{d}(n)$ defines a long-term prediction delay parameter for every time instant of the current frame. The adaptive codebook excitation $v(n) = u(n - \tilde{d}(n))$ is formed from the past excitation for the current subframe as in the encoder using the delay contour $\tilde{d}(n)$. The next section provides the details of the signal modification procedure as well as its use as part of the rate determination algorithm.

5.11.1 Search of Pitch Pulses and Pitch Cycle Segments

The signal modification scheme synchronously operates on pitch and frame by shifting each detected pitch cycle segment individually while constraining the shift at frame boundaries. This requires a mechanism for locating pitch pulses and corresponding pitch cycle segments for the current frame. Pitch cycle segments are determined based on detected pitch pulses that are searched according to Figure 5.11-1.

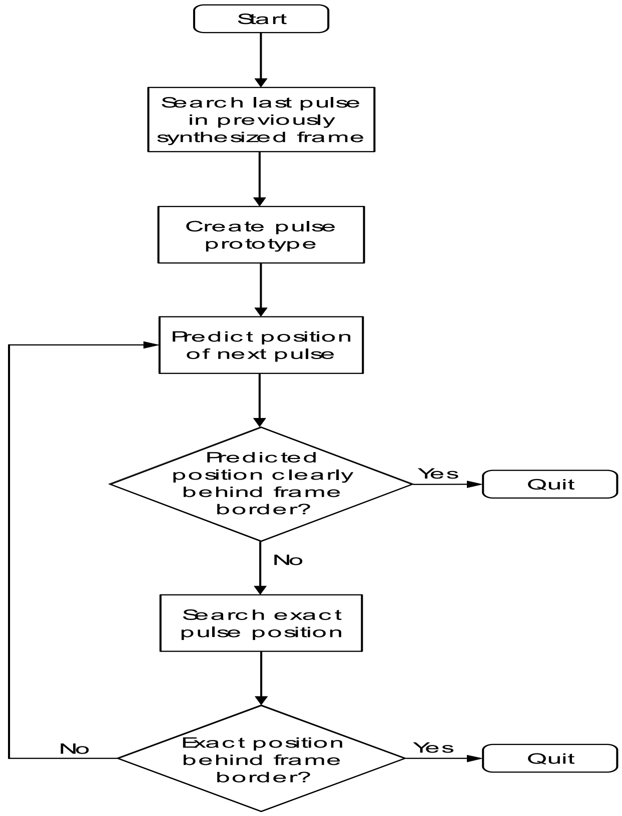

Figure 5.11-1: Functional block diagram of pitch pulse search.

Pitch pulse search operates on the weighted speech signal $s_w(n)$ and the weighted synthesized speech signal $\hat{s}_w(n)$. It should be noted that the weighted speech signal $s_w(n)$ is needed also for the lookahead in order to search the last pitch pulse in the current frame. This is done by using the weighting filter formed in the last subframe of the current frame over the lookahead portion.

The pitch pulse search procedure of Figure 5.11-1 starts by locating the last pitch pulse of the previous frame from the residual signal $r(n)$. A pitch pulse typically can be clearly distinguished as the maximum absolute value of the low-pass filtered residual signal in a pitch cycle having a length of approximately p_{k-1}, where p_{k-1} is the estimated pitch value at the frame end ($p_{k-1} = d_{-1}$) in case of successful signal modification in the previous frame, or the closed-loop pitch estimate for the last subframe of the previous frame ($p_{k-1} = \bar{d}_{k-1}$). A normalized Hamming window $H_5(z) = (0.08\,z^{-2} + 0.54\,z^{-1} + 1 + 0.54\,z + 0.08\,z^2)/2.24$ having a length of five samples is used for the low-pass filtering in order to efficiently locate the last pitch pulse of the previous frame. This pitch pulse position is denoted by T_0. The signal modification method does not require an accurate position for this pitch pulse, but rather a rough location estimate of the high-energy segment in the pitch cycle.

After locating the last pitch pulse at T_0 in the previous frame, a pitch pulse prototype of length 21 samples is extracted around this rough position estimate as

$$m_n(i) = \hat{s}_w(T_0 - 10 + i) \quad \text{for } i = 0, 1, \ldots, 20. \qquad (5.11.1\text{-}1)$$

This pitch pulse prototype is subsequently used in locating pitch pulses in the current frame.

The synthesized weighted speech signal $\hat{s}_w(n)$ is used for the pulse prototype instead of the residual signal $r(n)$. This facilitates pitch pulse search because the periodic structure of the signal is better preserved in the weighted speech signal. The synthesized weighted speech signal $\hat{s}_w(n)$ is obtained by filtering the synthesized speech signal $\hat{s}(n)$ of the previous frame by the weighting filter $W(z)$. If the pitch pulse prototype extends over the end of the previously synthesized frame, the weighted speech signal $s_w(n)$ of the current frame is used for this exceeding portion. The pitch pulse prototype has a high correlation with the pitch pulses of the weighted speech signal $s_w(n)$ if the previous synthesized speech frame contains already a well-developed pitch cycle. Thus the use of the synthesized speech in extracting the prototype provides additional information for monitoring the performance of coding and for selecting an appropriate encoding method in the current frame as will be explained in more detail in the following description.

Given the position T_0 of the last pulse in the previous frame, the first pitch pulse of the current frame can be predicted to occur approximately at instant $T_0 + p(T_0)$. Here $p(t)$ denotes the interpolated open-loop pitch estimate at instant (position) t, that is

$$p(t) = p_{t0} + (p_{t1} - p_{t0})\frac{t}{128},$$

where

$p_{t0} = p_{k-1}$, $p_{t1} = d_0$ for $0 \le t < 127$

$p_{t0} = d_0$, $p_{t1} = d_1$ for $128 \le t < 255$.

$p_{t0} = d_1$, $p_{t1} = d_2$ for $256 \le t$.

p_{k-1} denotes the estimated pitch value at the frame end in case of successful signal modification in the previous frame, or the closed-loop pitch estimate for the last subframe of the previous frame, and d_0, d_1, d_2 are the open-loop pitch estimates for 2 half-frames of the current frame and the look-ahead. Note that t=0 corresponds to the first sample of the current frame.

The predicted pitch pulse position $T_0 + p(T_0)$ is then refined as

$$T_1 = T_0 + p(T_0) + \arg\max C(j), \tag{5.11.1-2}$$

where $C(j)$ is the weighted correlation between weighted speech signal $s_w(n)$ in the neighbourhood of the predicted position and the pulse prototype

$$C(j) = \gamma(j)\sum_{k=0}^{20} m_n(k)s_w(T_0 + p(T_0) + j - 10 + k), \qquad j \in [-j_{\max}, j_{\max}]. \tag{5.11.1-3}$$

Thus the refinement is the argument j, limited into $[-j_{\max}, j_{\max}]$, that maximizes the weighted correlation $C(j)$. The limit $j_{\max}$ is proportional to the open-loop pitch estimate as $\min\{20, \langle p(0)/4\rangle\}$, where the operator $\langle \cdot \rangle$ denotes rounding to the nearest integer. The weighting function

$$\gamma(j) = 1 - |j| / p(T_0 + p(T_0)) \tag{5.11.1-4}$$

Equation (5.11.1-3) favors the pulse position predicted using the open-loop pitch estimate, since $\gamma(j)$ attains its maximum value 1 at j = 0. The denominator $p(T_0 + p(T_0))$ in Equation (5.11.1-4) is the interpolated open-loop pitch estimate for the predicted pitch pulse position.

After the first pitch pulse position T_1 has been found using Equation (5.11.1-2), the next pitch pulse can be predicted to be at instant $T_2 = T_1 + p(T_1)$ and refined as described above. This pitch pulse search comprising the prediction and refinement is repeated until either the prediction or refinement procedure yields a pitch pulse position outside the current frame. It should be noted that the pitch pulse prediction terminates the search only if a predicted pulse position is extremely far in the subsequent frame that the refinement step cannot bring it back to the current frame. This procedure yields c pitch pulse positions inside the current frame, denoted by T_1, T_2, …, T_c.

Pitch pulses are located with the integer resolution except the last pitch pulse of the frame denoted by T_c. Since the exact distance between the last pulses of two successive frames is needed to determine the delay parameter to be transmitted, the last pulse position is refined with a fractional resolution of ¼ sample by maximizing the following correlation function for i, j.

$$C(j,i) = \sum_{i=-16}^{16} s_w(T_0 + j + i) s_w(T_c + i/4 + i), \quad j \in [-2, 1], i \in [0, 3] \qquad (5.11.1\text{-}5)$$

The fractional resolution is obtained by up-sampling the weighted speech signal $s_w(n)$ in the neighborhood of the last predicted pitch pulse before evaluating the correlation function. Hamming-windowed sinc interpolation of length 33 is used for up-sampling. The fractional resolution of the last pitch pulse position helps to maintain a good performance of the long-term prediction despite the time synchrony constraint set to the frame end.

After completing pitch cycle segmentation in the current frame, an optimal shift for each segment is calculated. This operation is done using the weighted speech signal $s_w(n)$ as will be explained in the following description. For reducing the distortion, the shifts of individual pitch cycle segments are implemented using the LP residual signal $r(n)$. Since shifting distorts the signal particularly around segment boundaries, it is essential to place the boundaries in low power sections of the residual signal $r(n)$. The segment boundaries are placed approximately in the middle of two consecutive pitch pulses, but constrained inside the current frame. Segment boundaries are always selected inside the current frame such that each segment contains exactly one pitch pulse. Segments with more than one pitch pulse or "empty" segments without any pitch pulses restrain subsequent correlation-based matching with the target signal and should be prevented in pitch cycle segmentation. The sth extracted segment of l_s samples is denoted as $w_s(i)$ for i = 0, 1, …, l_s - 1. The starting instant of this segment is t_s, selected such that $w_s(0) = s_w(t_s)$. The number of segments in the present frame is denoted by c.

While selecting the segment boundary between two successive pitch pulses T_s and T_{s+1} inside the current frame, the following procedure is used. First the central instant between two pulses is computed as $\Lambda = \langle (T_s + T_{s+1})/2 \rangle$. The candidate positions for the segment boundary are located in the region $[\Lambda - \varepsilon_{\max}, \Lambda + \varepsilon_{\max}]$, where $\varepsilon_{\max}$ corresponds to five samples. The energy of each candidate boundary position is computed as

$$Q(\varepsilon') = 0.75 r^2(\Lambda + \varepsilon' - 1) + r^2(\Lambda + \varepsilon') + 0.75 r^2(\Lambda + \varepsilon' - 1), \qquad \varepsilon' \in [-\varepsilon_{\max}, \varepsilon_{\max}] \qquad (5.11.1\text{-}5)$$

The position corresponding to the smallest energy is selected because this choice typically results in the smallest distortion in the modified speech signal. The instant that minimizes Equation (5.11.1-5) is denoted as ε. The starting instant of the new segment is selected as $t_s = \Lambda + \varepsilon + 1$. This defines also the length of the previous segment, since the previous segment ends at instant $\Lambda + \varepsilon$.

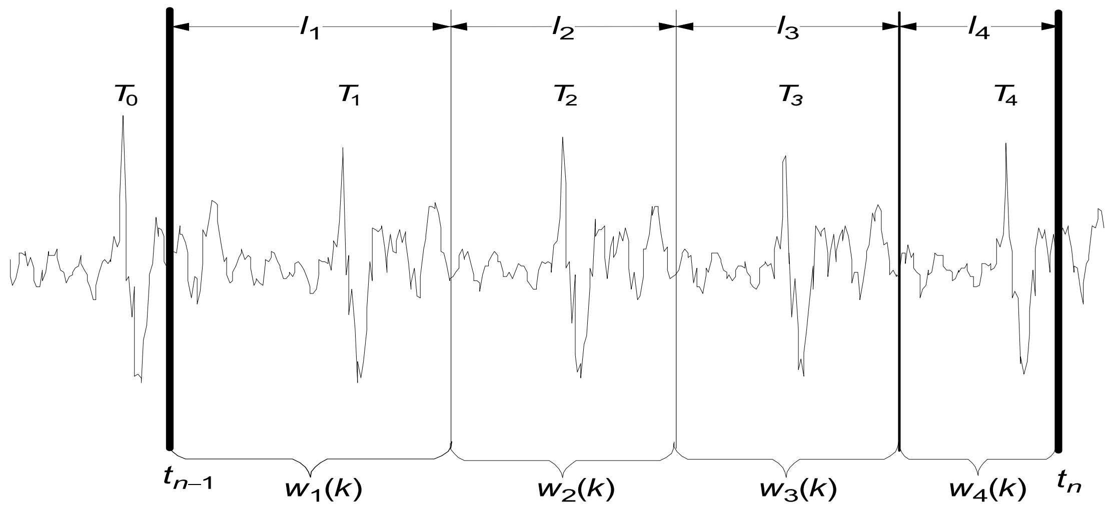

Figure 5.11-2: Illustrative example on located pitch pulse positions and the corresponding pitch cycle segmentation for one frame

5.11.2 Determination of the Delay Parameter

The main advantage of signal modification is that only one delay parameter per frame has to be encoded and transmitted to the decoder. However, special attention has to be paid to the determination of this single parameter. The delay parameter $\bar{d}_k$ not only defines together with its previous value the evolution of the pitch cycle length over the frame, but also affects the time synchrony in the resulting modified signal.

In TIA/EIA/IS-127 codec [8, 20], no time synchrony is required at frame boundaries, and thus the delay parameter to be transmitted can be directly determined using an open-loop pitch estimate. This selection usually results in a time asynchrony at the frame boundary and translates into an accumulating time shift in the subsequent frame because the signal continuity has to be preserved. In the VMR-WB codec, the signal modification method preserves the time synchrony at frame boundaries. Thus, a strictly constrained shift occurs at the frame boundaries and every new frame starts in perfect time match with the original speech frame.

To ensure time synchrony at the frame boundaries, the delay contour $\tilde{d}(n)$ maps, with the long-term prediction, the last pitch pulse at the end of the previous synthesized speech frame to the last pitch pulse of the current frame. The delay contour defines an interpolated long-term prediction delay parameter over the current kth frame for every sample from instant t_{k-1} + 1 through t_k (0 to L-1). Only the delay parameter $\bar{d}_k = \tilde{d}(t_k) = \tilde{d}(L-1)$ at the frame end is transmitted to the decoder implying that $\tilde{d}(n)$ must have a form fully specified by the transmitted values. The long-term prediction delay parameter has to be selected such that the resulting delay contour fulfills the pulse mapping. In a mathematical form, this mapping can be presented as follows: Let κ_c be a temporary time variable and T_0 and T_c the last pitch pulse positions in the previous and the current frames, respectively. The delay parameter $\bar{d}_k$ has to be selected such that, after executing the pseudo-code presented in Table 5.11-1, the variable κ_c has a value very close to T_0 minimizing the error $e_n = |\kappa_c - T_0|$. The pseudo-code starts from the value $\kappa_0 = T_c$ and iterates backwards c times by updating $\kappa_i := \kappa_{i-1} - \tilde{d}(\kappa_{i-1})$. If κ_c then equals to T_0, long-term prediction can be utilized with maximum efficiency without time asynchrony at the frame end. In practice, the tolerated error e_n is limited to 1.

Table 5.11-1: Loop for searching the optimal delay parameter

% Initialization
$\kappa_0 := T_c;$

% Loop
for i = 1 to c
$\quad \kappa_i := \kappa_{i-1} - \tilde{d}\,(\kappa_{i-1})\,;$
end;

An example of the operation of the delay selection loop in the case c = 3 is illustrated in Figure 5.11-3. The loop starts from the value $\kappa_0 = T_c$ and takes the first iteration backwards as $\kappa_1 = \kappa_0 - \tilde{d}\,(\kappa_0)$. Iterations are continued twice more resulting in $\kappa_2 = \kappa_1 - \tilde{d}\,(\kappa_1)$ and $\kappa_3 = \kappa_2 - \tilde{d}\,(\kappa_2)$. The final value κ_3 is then compared against T_0 in terms of the error $e_n = |\kappa_3 - T_0|$. The resulting error is a function of the delay contour that is computed in the delay selection algorithm as will be shown later in this specification.

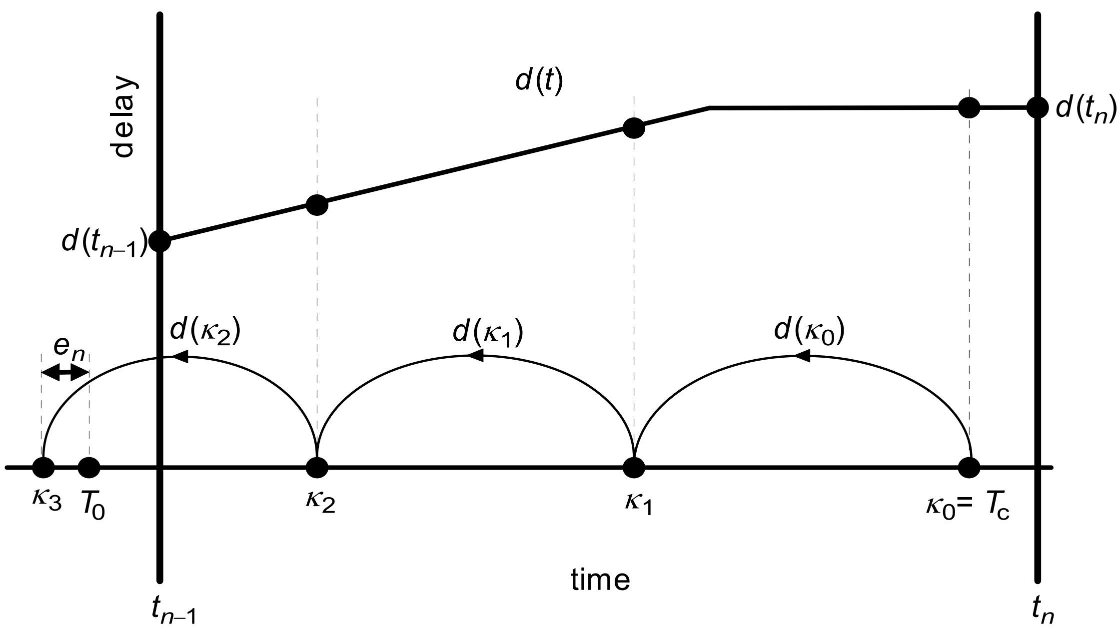

Figure 5.11-3: Illustrative example on the operation of the delay selection procedure when the number of pitch pulses is three (c = 3).

The signal modification method in TIA/EIA/IS-127 codec [8] interpolates the delay parameters linearly over the frame between $\bar{d}_{k-1}$ and $\bar{d}_k$. However, when time synchrony is required at the frame end, linear interpolation tends to result in an oscillating delay contour. Thus pitch cycles in the modified speech signal contract and expand periodically causing annoying audible artifacts. The evolution and amplitude of the oscillations are related to the last pitch position. The farther the last pitch pulse is from the frame end in relation to the pitch period, the more likely the oscillations are amplified. Therefore, a piecewise linear delay contour is used as follows:

$$\tilde{d}(n) = \begin{cases} (1-\alpha(n))\bar{d}_{k-1} + \alpha(n)\bar{d}_k, & t_{k-1} < n \le t_{k-1} + \sigma_k \\ \bar{d}_k, & t_{k-1} + \sigma_k < n < t_k \end{cases} \tag{5.11.2-1}$$

where

$$\alpha(n) = (n)/\sigma_k \tag{5.11.2-2}$$

The oscillations are significantly reduced by using this delay contour. Here t_n and t_{n-1} are the end instants of the current and previous frames, respectively, and $\bar{d}_k$ and $\bar{d}_{k-1}$ are the corresponding delay parameter values. Note that $t_{k-1} + \sigma_k$ (σ_k -1) is the instant after which the delay contour remains constant. The parameter σ_k varies as a function of $\bar{d}_{k-1}$ as

$$\sigma_k = \begin{cases} 172 \text{ samples}, & \bar{d}_{k-1} \le 90 \text{ samples} \\ 128 \text{ samples}, & \bar{d}_{k-1} > 90 \text{ samples} \end{cases} \tag{5.11.2-3}$$

and the frame length L is 256 samples. To avoid oscillations, it is beneficial to decrease the value of σ_k as the length of the pitch cycle increases. On the other hand, to avoid rapid changes in the delay contour $\tilde{d}(n)$ in the beginning of the frame ($n < \sigma_k$), the parameter σ_k has to be always at least a half of the frame length. Rapid changes in $\tilde{d}(n)$ degrade easily the quality of the modified speech signal.

Note that depending on the encoding method of the previous frame $\bar{d}_{k-1}$ can be either the delay value at the frame end (signal modification enabled) or the close-loop delay value of the last subframe (signal modification disabled). Since the past value $\bar{d}_{k-1}$ of the delay parameter is known at the decoder, the delay contour is unambiguously defined by $\bar{d}_k$, and the decoder is able to form the delay contour using Equation (5.11.2-1).

The only parameter, which can be varied while searching the optimal delay contour, is $\bar{d}_k$, the delay parameter value at the end of the frame constrained into [34, 231] samples. There is no simple explicit method for solving the optimal $\bar{d}_k$ in a general case. Instead, several values have to be tested to find the best solution. However, the search is straightforward. The value of $\bar{d}_k$ is first predicted as

$$\bar{d}_k^{(0)} = 2\frac{T_c - T_0}{c} - \bar{d}_{k-1} - 2. \tag{5.11.2-4}$$

Then, the search is conducted in three phases by increasing the resolution and focusing the search range to be examined inside [34, 231] in every phase. The delay parameters giving the smallest error $e_n = |\kappa_c - T_0|$ in the procedure of Table 5.11-1 in these three phases are denoted by $\bar{d}_k^{(1)}$, $\bar{d}_k^{(2)}$, and $\bar{d}_k = \bar{d}_k^{(3)}$, respectively. In the first phase, the search is done around the value $\bar{d}_k^{(0)}$ predicted using Equation (5.11.2-4) with a resolution of 4 samples in the range [$\bar{d}_k^{(0)}$ – 11, $\bar{d}_k^{(0)}$ + 12] when $\bar{d}_k^{(0)}$ < 60, and in the range [$\bar{d}_k^{(0)}$ – 15, $\bar{d}_k^{(0)}$ + 16] otherwise. The second phase constrains the range into [$\bar{d}_k^{(1)}$ – 3, $\bar{d}_k^{(1)}$ + 3] and uses the integer resolution. The last, third phase examines the range [$\bar{d}_k^{(2)}$ – 3/4, $\bar{d}_k^{(2)}$ + 3/4] with a resolution of ¼ sample for $\bar{d}_k^{(2)}$ < $92^1/_2$. Above that range [$\bar{d}_k^{(2)}$ – 1/2,

$\bar{d}_k^{(2)}$ + 1/2] and a resolution of ½ sample is used. This third phase yields the optimal delay parameter $\bar{d}_k$ to be transmitted to the decoder. This procedure is a compromise between the search accuracy and complexity.

The delay parameter $\bar{d}_k \in$ [34, 231] can be coded using nine bits per frame using a resolution of 1/4 sample for $\bar{d}_k < 92^1/_2$ and ½ sample for $\bar{d}_k > 92^1/_2$.

5.11.3 Modification of the Signal

After the delay parameter $\bar{d}_k$ and the pitch cycle segmentation have been determined, the signal modification procedure itself can be initiated. The speech signal is modified by shifting individual pitch cycle segments one by one adjusting them to the delay contour $\tilde{d}(n)$. A segment shift is determined by correlating the segment in the weighted speech domain with a target signal. The target signal is composed using the synthesized weighted speech signal $\hat{s}_w(n)$ of the previous frame and the preceding, already shifted segments in the current frame, together with the delay contour $\tilde{d}(n)$. The actual shift is done on the residual signal $r(n)$.

Signal modification has to be done carefully to both maximize the performance of long-term prediction and simultaneously to preserve the perceptual quality of the modified speech signal. The required time synchrony at frame boundaries has to be taken into account also during modification.

A block diagram of the signal modification method is shown in Figure 5.11-4. Modification starts by extracting a new segment $w_s(n)$ of l_s samples from the weighted speech signal $s_w(n)$. This segment is defined by the segment length l_s and starting instant t_s giving $w_s(n) = w(t_s + n)$ for $k = 0, 1, \ldots, l_s - 1$. The segmentation procedure shall be performed in accordance with the foregoing description.

For finding the optimal shift of the current segment $w_s(n)$, a target signal $\tilde{w}(n)$ is created. For the first segment $w_1(k)$ in the current frame, this target signal is obtained by the recursion

$$\begin{aligned} \tilde{w}(n) &= \hat{s}_w(n), & n &< 0 \\ \tilde{w}(n) &= \tilde{w}(n - \tilde{d}(n)), & n &= 0, \ldots, l_1 + \delta_s - 1. \end{aligned} \qquad (5.11.3\text{-}1)$$

Here $\hat{s}_w(n)$ is the weighted synthesized speech signal available in the previous frame for $n \le t_{k-1}$. The parameter $\delta_s = 5$ is the range of the shift search and l_1 is the length of the first segment. Equation (5.11.3-1) can be interpreted as simulation of long-term prediction using the delay contour over the signal portion in which the current shifted segment may potentially be situated. The computation of the target signal for the subsequent segments follows the same principle and will be presented later in this section.

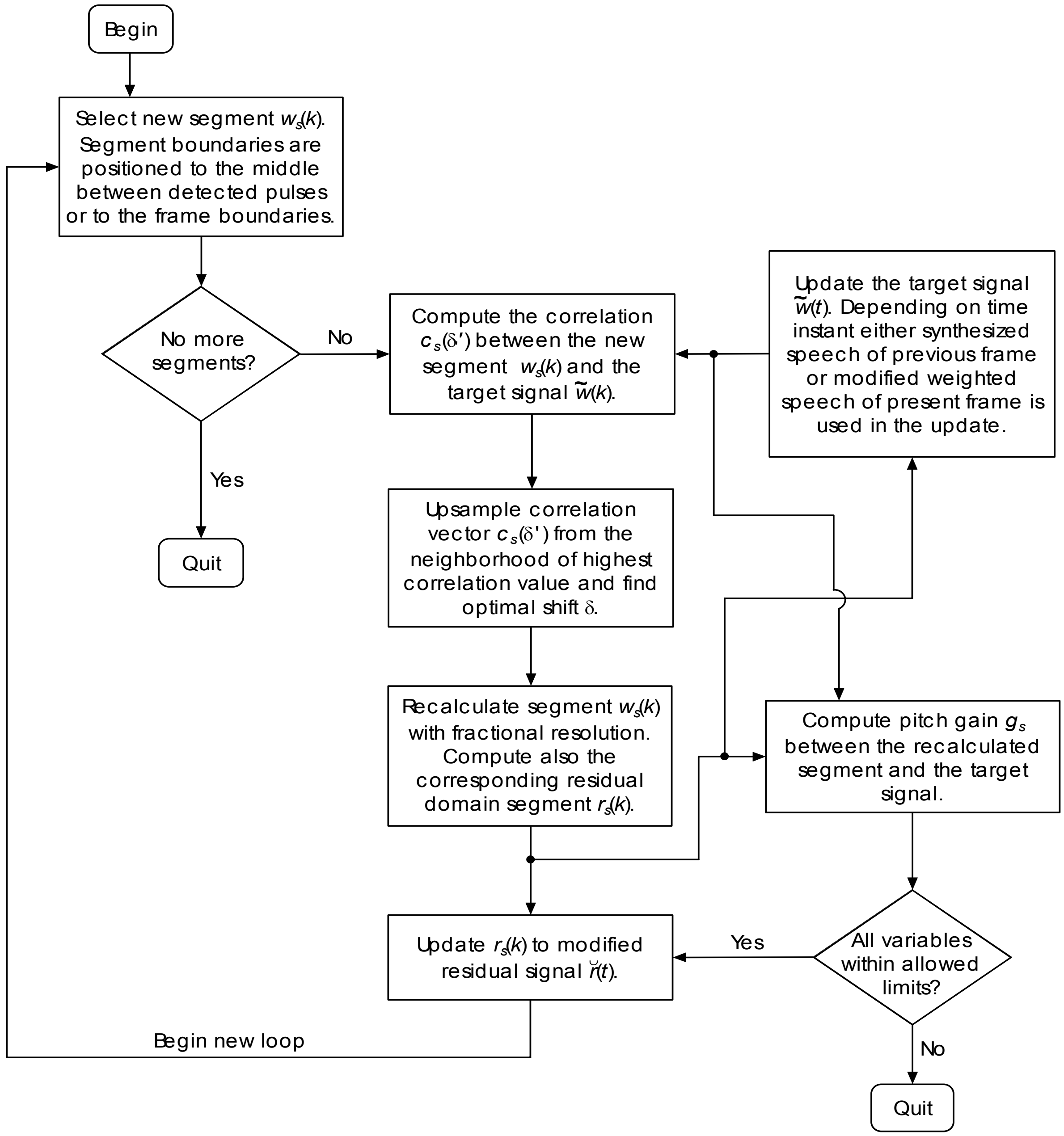

Figure 5.11-4: Functional block diagram on the signal modification procedure that adjusts the speech frame to the selected delay contour

The search procedure for finding the optimal shift of the current segment can be initiated after forming the target signal. This procedure is based on the correlation $c_s(\delta')$ between the segment $w_s(n)$ that starts at instant t_s and the target signal $\tilde{w}(n)$ as

$$c_s(\delta') = \sum_{k=0}^{l_s-1} w_s(k)\tilde{w}(k + t_s + \delta'),\ \delta' \in [-\delta_s, \delta_s], \qquad (5.11.3\text{-}2)$$

After the integer shift δ that maximizes the correlation $c_s(\delta')$ in (5.11.3-2) is found, the maximum correlation value is searched with a fractional resolution in the open interval $(\delta - 1, \delta + 1)$, and bounded into [-δ_s, δ_s]. The correlation $c_s(\delta')$ is up-sampled in this interval to a resolution of 1/8 sample using Hamming-windowed sinc interpolation of a length equal to 65 samples. The shift δ corresponding to the maximum value of the up-sampled correlation is then the optimal shift with a fractional resolution. After finding this optimal shift, its fractional part is incorporated into the weighted speech segment $w_s(n)$. That is, the precise new starting instant of the segment is updated as $t_s := t_s - \delta + \delta_I$, where $\delta_I = \lceil \delta \rceil$ is the upward-rounded shift. Similarly, the fractional part of the optimal shift is also incorporated into the residual segment $r_s(n)$ corresponding to the weighted speech segment $w_s(n)$ and computed from the residual signal $r(n)$ using again the sinc interpolation as described before. Since the fractional part of the optimal shift is now incorporated into the residual and weighted speech segments, all subsequent computations can be implemented with $\delta_I = \lceil \delta \rceil$.

Even if the optimal shift search range is [-5,5] samples, the actual maximum permitted shift varies with the pitch period. The following values are used for δ_s:

$$\delta_s = \begin{cases} 4\,{}^1\!/_2 \text{ samples}, & \bar{d}_k < 90 \text{ samples} \\ 5 \text{ samples}, & \bar{d}_k \geq 90 \text{ samples} \end{cases} \tag{5.11.3-3}$$

If the correlation function is maximized for $\delta > \delta_s$, the signal modification is aborted and Voiced HR encoding type is not used. As will be described later in this section, the value of δ_s is even more limited for the first and the last segment in the frame.

The final task is to update the modified residual signal $\breve{r}(n)$ by copying the current residual signal segment $r_s(n)$ into it:

$$\breve{r}(t_s + \delta_I + n) = r_s(n), \quad n = 0, 1, ..., l_s - 1 \tag{5.11.3-4}$$

Since shifts in successive segments are independent from each other, the segments positioned to $\breve{r}(n)$ either overlap or have a gap in between them. Straightforward weighted averaging is used for overlapping segments, that is for negative δ_I

$$\breve{r}(t_s + \delta_I + k) = \left(1 - \frac{k+1}{|\delta_I|}\right) \breve{r}(t_s + \delta_I + k) + \left(\frac{k+1}{|\delta_I|}\right) r_s(k), \quad k = 0, ..., |\delta_I| - 1 \tag{5.11.3-5}$$

The remaining segment samples are simply copied following (5.11.3-4). The gaps are filled with zeros. There are only 2 exceptions. The first exception is the case of the first segment when only one sample is missing, that is the fractional shift δ lower than 1 has been selected. In this case the residual signal $r(n)$ is up-sampled with 1/8 sample resolution around the instant t_{k-1} and the gap is filled with the sample closest to $r(t_{k-1} + (\delta_I - \delta)/2)$. The second exception is when some samples are missing at the end of the last segment. In this case, the last available sample is simply repeated until the frame ends. Since the number of overlapping or missing samples is usually small and the segment boundaries occur at low-energy regions of the residual signal, usually no perceptual artefacts are caused.

Processing of the subsequent pitch cycle segments follows the above-mentioned procedure, except that the target signal $\tilde{w}(t)$ is formed differently than for the first segment. The samples of $\tilde{w}(n)$ are first replaced with the modified weighted speech samples as

$$\tilde{w}(t_s + \delta_I + n) = w_s(n), \quad n = 0, 1, ..., l_s - 1 \tag{5.11.3-6}$$

When updating the target signal following (5.11.3-6), the segments can either overlap or have a gap in between them similarly as explained above for the residual signal. Overlapping segments in the target signal are processed exactly the same way as in the modified residual signal. If samples are missing (positive δ_I), a linear interpolation is used between the last available sample in $\tilde{w}(n)$ and the first sample of $w_s(n)$

$$\tilde{w}(t_s+k) = (1-\frac{k+1}{|\delta_I|})\tilde{w}(t_s-1) \quad + \quad (\frac{k+1}{|\delta_I|})w_s(0), \qquad k=0,...,|\delta_I|-1 \qquad (5.11.3\text{-}7)$$

Again, the case of the first segment with only one missing sample is processed differently. The missing sample is simply the average of $\tilde{w}(t_{n-1})$ (the last sample of the previous frame) and $w_s(0)$. Then the samples following the updated segment are also updated,

$$\tilde{w}(n) \quad = \quad \tilde{w}(n-\tilde{d}(k)), \qquad n \; = \; t_s+\delta_I+l_s, \;\; ..., \;\; t_s+\delta_I+l_s+l_{s+1}+4 \qquad (5.11.3\text{-}8)$$

The update of the target signal $\tilde{w}(n)$ ensures higher correlation between successive pitch cycle segments in the modified speech signal following the delay contour $\tilde{d}(n)$ and thus more accurate long-term prediction. While processing the last segment of the frame, the target signal $\tilde{w}(n)$ does not need to be updated.

The shifts of the first and the last segments in the frame are special cases. Before shifting the first segment, it should be ensured that no high power regions exist in the residual signal $r(n)$ close to the frame boundary t_{k-1}, because shifting such a segment may cause artifacts. The high power region is searched by squaring the first pitch period of the residual signal $r(n)$ as

$$E_0(i) = r^2(i), \qquad j = 0,...,\bar{d}_{k-1}-1 \qquad (5.11.3\text{-}9)$$

where $\bar{d}_{k-1}$ is the pitch period at the end of the previous frame as described in Section 5.11.2. If the maximum of $E_0(i)$ is detected close to the frame boundary in the range $[t_{k-1}+1,\ t_{k-1}+2]$, the allowed shift is limited to ¼ samples. The allowed shift is limited to ¼ samples also if one of the pitch pulse positions T_0 or T_1 as described in Section 5.11.1 are found in the range $[t_{k-1},\ t_{k-1}+2]$. If the proposed shift $|\delta|$ for the first segment is smaller that this limit, the signal modification procedure is enabled in the current frame, but the first segment is kept intact.

The last segment in the frame is processed in a similar manner. As was described in the foregoing description, the delay contour $\tilde{d}(n)$ is selected such that in principle no shifts are required for the last segment. However, because the target signal is repeatedly updated during signal modification, it is possible the last segment has to be shifted slightly. This shift is always constrained to be smaller than 3/2 samples. The shift limit is further reduced to be smaller than 0.5 samples if the last pulse position T_c is close to the frame boundary, that is

$$T_c > t_k+1 - \min\{10,\ 0.22 * \bar{d}_k\}$$

If the signal modification algorithm suggests larger shift than permitted, signal modification is aborted and Generic FR encoding type is used.

If there is a high power region at the frame end, no shift is allowed at all. This condition is verified by using the squared residual signal $r^2(i)$. If the maximum of $r^2(i)$ evaluated for $i \in [t_k+2-\bar{d}_k,\ t_k+2]$ is attained for i larger than t_k or if the maximum of $r^2(i)$ evaluated for $i \in [t_k+7-\bar{d}_k,\ t_k+6]$ is attained for i larger than t_k-4, no shift is allowed for the last segment. Similarly as for the first segment, when the proposed shift $|\delta| < ¼$, the present frame is still accepted for modification, but the last segment is kept intact.

It should be noted that the shift does not translate into the next frame, and every new frame starts perfectly synchronized with the original input signal.

If the signal modification procedure is successful and after determining the modified residual signal, the modified speech signal $s(n)$ is computed by filtering the residual signal through the synthesis filter 1/A(z). The modified signal $s(n)$ replaces the input denoised signal in the subsequent processing for computing the adaptive and algebraic codebook contributions.

5.11.4 Voiced Classification Logic Incorporated into the Signal Modification Procedure

The signal modification method incorporates an efficient classification and mode determination mechanism as depicted in Figure 5.11-5. Every operation yields several indicators quantifying the attainable performance of long-term prediction in the current frame. If any of these indicators are outside their allowed limits, the signal modification procedure is terminated and the original signal is preserved intact.

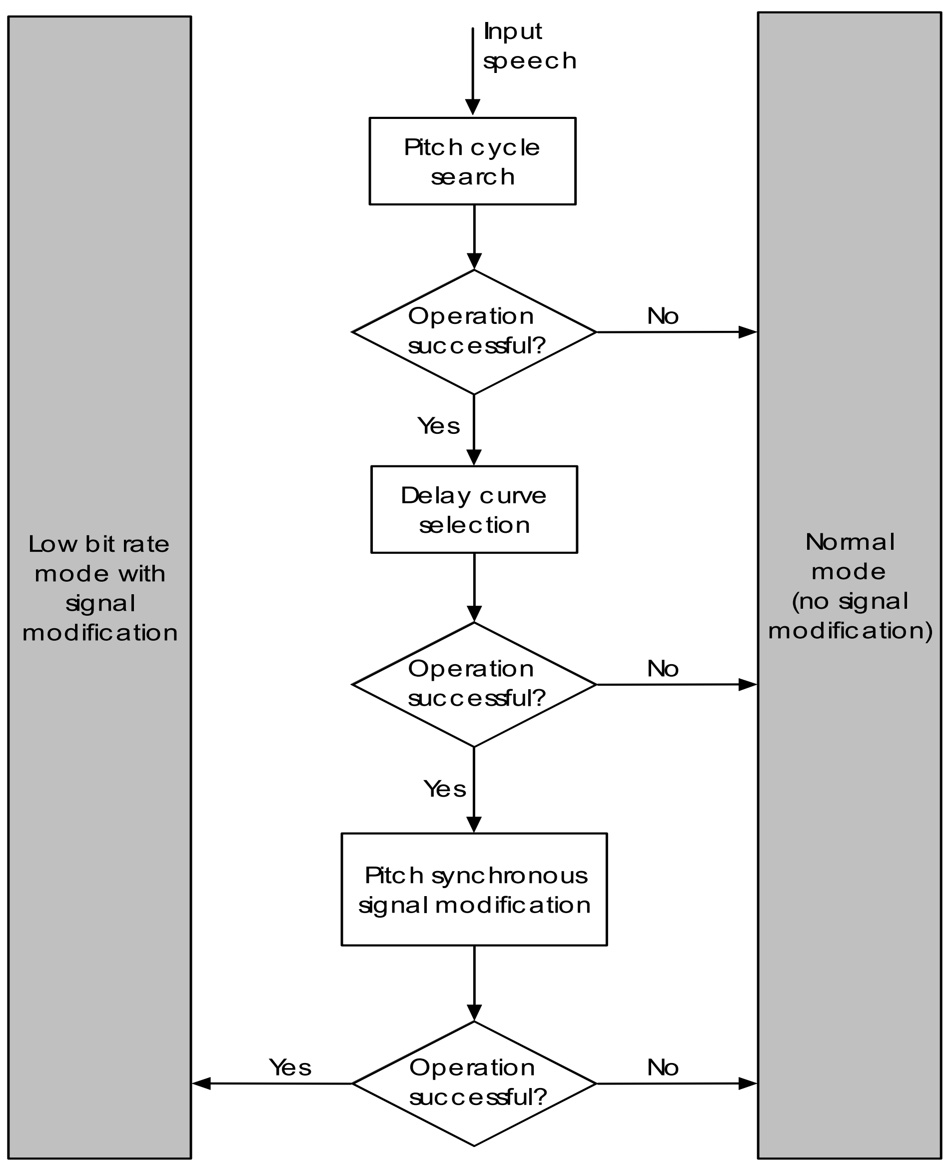

Figure 5.11-5: Functional block diagram of closed loop voiced signal classification.

The pitch pulse search procedure produces several indicators on the periodicity of the present frame. First, the signal modification is aborted immediately if the difference between the pitch at the end of the previous frame and the open-loop pitch estimate at the end of the current frame is significant, that is if

$$\left|4[\bar{d}_{k-1} - p(t_k)]\right| > \bar{d}_{k-1}, \qquad \text{Or}$$

$$\left|4[\bar{d}_{k-1} - p(t_k)]\right| > p(t_k).$$

$p(t_k)$ is the interpolated open loop pitch estimate for the last sample t_k in the current frame given by $p(t_k) = 0.9921875\, d_0 + 0.0078125\, d_1$ where d_0 and d_1 are the open loop pitch estimates for the first and the second half-frame of the current frame, respectively (Section 5.8).

Second, if the difference between the detected pitch pulse positions and the interpolated open-loop pitch estimate is not within a permitted range, that is

$$|0.4[T_i - T_{i-1} - p(T_i)]| > 0.2\, p(T_i), \quad i = 1, 2, \ldots, c, \tag{5.11.4-1}$$

the signal modification procedure is aborted.

The selection of the delay contour $\tilde{d}(n)$ provides additional information on the evolution of the pitch cycles and the periodicity of the current speech frame. The signal modification procedure is continued only if the condition $|\bar{d}_k - \bar{d}_{k-1}| \le \delta_d\, \bar{d}_k$ is satisfied. In VMR-WB mode 2, δ_d varies linearly with the relative frame energy E_{rel} as

$$\delta_d = -0.0429 * E_{rel} + 0.0714 \tag{5.11.4-2}$$

and is constrained between 0.2 and 0.5; otherwise δ_d equals 0.2. This condition means that only a small delay change is tolerated for classifying the current frame as purely voiced frame. The algorithm also evaluates the success of the delay selection loop of Table 5.11-1 by examining the difference $|\kappa_c - T_0|$ for the selected delay parameter value $\bar{d}_k$. If this difference is greater than one sample, the signal modification procedure is terminated.

To ensure a good quality for the modified speech signal, it is advantageous to constrain the shifts performed for successive pitch cycle segments. This is achieved by imposing the criteria

$$\left|\delta^{(s)} - \delta^{(s-1)}\right| \le \begin{cases} 4.0 \quad \text{samples}, & \bar{d}_k \le 90 \text{ samples} \\ 4.8 \quad \text{samples}, & \bar{d}_k > 90 \text{ samples} \end{cases} \tag{5.11.4-3}$$

on all segments of the frame. Here $\delta^{(s)}$ and $\delta^{(s-1)}$ are the shifts done for the sth and $(s-1)$th pitch cycle segments, respectively. If the thresholds are exceeded, the signal modification procedure is interrupted and the original signal is maintained.

When the frames subjected to signal modification are encoded at a low bit rate, it is essential that the shape of pitch cycle segments remains similar over the frame. This allows faithful signal modeling by long-term prediction and thus encoding at a low bit rate without degrading the subjective quality. The similarity of successive segments can be quantified simply by the normalized correlation

$$g_s = \frac{\sum_{k=0}^{l_s-1} w_s(k)\widetilde{w}(k+t_s+\delta_I)}{\sqrt{\sum_{k=0}^{l_s-1} w_s^2(k) \sum_{k=0}^{l_s-1} \widetilde{w}^2(k+t_s+\delta_I)}} \quad (5.11.4\text{-}4)$$

between the current segment $w_s(k)$ updated with the fractional part of the optimal shift, and the target signal $\widetilde{w}(n)$ at the optimal shift δ_I. The success of the procedure is examined using the criteria $g_s \geq \delta_g$. The threshold δ_g varies with the operational mode and the estimated pitch period. The reason is that the signal modification algorithm performs generally better for longer pitch periods.

If maximum half-rate is enforced for the current frame, δ_g = 0.5. In VMR-WB mode 2, δ_g varies linearly with E_{rel} to increase the Voiced HR usage in low energy frames: For d_n > 96, $\delta_g = 0.45 * E_{rel} + 0.79$ and is upper-bounded to 0.79. For $\overline{d}_k \leq 96$, $\delta_g = 0.05 * E_{rel} + 0.847$, upper bounded to 0.847. In VMR-WB mode 1, δ_g = 0.91 for $\overline{d}_k > 100$ and δ_g = 0.948 for $\overline{d}_k \leq 100$. The threshold δ_g has been used to adjust the selection of the Voiced HR encoding type to comply with the desired average bit rates. If the condition $g_s \geq \delta_g$ is not satisfied for all segments, the signal modification procedure is terminated and the original signal is kept intact.

If the signal modification is successfully performed, long-term prediction is able to model the modified speech frame efficiently, facilitating its encoding at a low bit rate without degrading subjective quality. In this case, the adaptive codebook excitation has a dominant contribution in describing the excitation signal, and thus the bit rate allocated for the fixed-codebook excitation can be reduced. Otherwise, the frame is likely to contain a non-stationary speech segment such as a voiced onset or rapidly evolving voiced speech signal. These frames typically require a high bit rate for maintaining good subjective quality.

5.12 Selection of FR and Generic HR, and Maximum and Minimum Rate Operation

Routine Name: `rate_select2`
Inputs:

- Mode of operation and HR-maximum signaling flag
- E_{rel}: Relative frame energy
- $\overline{N}_f$: Long-term average noise energy
- Open-loop lag in the first half-frame
- Previous frame encoding type

Outputs:

- Encoding type

Initialization:

- None

If the rate selection reaches this stage, then the frame has not been declared as inactive speech frame (not encoded with CNG-ER or CNG-QR), nor declared as unvoiced frame (not encoded with Unvoiced HR or Unvoiced QR), nor declared as stable voiced frame (not encoded with Voiced HR). Thus the frame is likely to contain a non-stationary speech segment such as a voiced onset or rapidly evolving voiced speech signal. These frames typically require a general-purpose encoding model at high bit rate for maintaining good speech quality. Thus in this case an appropriate FR encoding type is mainly used. However, frames with very low energy and not detected as non-speech, unvoiced or stable voiced can be encoded using Generic HR encoding in order to reduce the average data rate. In the AMR-WB interoperable mode, Generic HR encoding type is not used and all frames at this stage are encoded with interoperable FR in normal operation. In VMR-WB mode 0, Generic HR encoding type is not used in normal operation and all frames at this stage are encoded with Generic FR. In VMR-WB modes 1 and 2, Generic HR encoding is used if the relative energy E_{rel} is lower than a certain threshold th_{14}.

In VMR-WB mode 1, the threshold is given as follows:

If last frame type is Unvoiced HR then $th_{14} = -14$ *else* $th_{14} = -11$.

Furthermore, if the long-term average noise energy $\overline{N}_f$ is larger than 21 then th_{14} is incremented by 2.7. In VMR-WB mode 2, the threshold is given as follows:

If the pitch open-loop lag in the first half-frame is larger than 100 then $th_{14} = -6$ *else* $th_{14} = -5$.

5.12.1 Maximum and Minimum Rate Operation

Depending on the application, a maximum or a minimum bit rate for a particular frame can be forced. Most often, the maximum bit rate imposed by the system during signaling periods (e.g., dim and burst) is limited to HR. However, the system can impose also lower rates. In normal operation and by default the maximum rate is FR and the minimum rate is ER.

If minimum and maximum rates are different from the default values then the signal classification described above will take this into account while making the rate selection.

For minimum rate operation, if min-rate is FR then lower rates cannot be used and all frames are encoded with an appropriate FR encoding type.

If min-rate is HR then QR and ER cannot be used. In this case, in AMR-WB interoperable mode, inactive speech frames are encoded with Interoperable HR and in other modes inactive frames are encoded with Unvoiced HR. The rest of classification is as usual.

If min-rate is QR then in all modes, all inactive frames are encoded with CNG-QR. The rest of classification is as usual.

If min-rate is ER, then the rate selection works in normal operation. In this case in the AMR-WB interoperable mode, the first inactive (VAD_flag=0) speech frame is encoded with CNG-QR (corresponding to SID_UPDATE in AMR-WB) then the next *k*-1 frames encoded with CNG-ER (those frames would be discarded at system interface with AMR-WB), then the *k*th frame is encoded with CNG-QR and so on, until active speech frames resume. In the VMR-WB codec, k is set to 8, that is CNG-QR is used every 8th frame. In the other modes, all inactive frames are encoded with CNG-ER.

In the case of maximum HR limitation, all active speech frames that would be classified as FR during normal operation must be encoded using an appropriate HR encoding type. The classification and rate selection mechanism are preformed as usual. That is determining inactive speech frames (encoded with CNG-ER or CNG-QR), unvoiced frames (encoded with Unvoiced HR or QR), stable-voiced frames (encoded with Voiced HR). All remaining frames that would be classified as FR during normal operation are encoded using the Generic HR encoding type except in the AMR-WB interoperable mode where an interoperable HR encoding type is used.

In the AMR-WB interoperable mode, the interoperable HR encoding type was designed to enable interoperable interconnection between VMR-WB and AMR-WB in case of maximum half-rate operation with minimal impact on the quality. Since there is no AMR-WB codec mode at 6.2 kbps or below, the interoperable HR was designed by dropping some of the bits corresponding to the algebraic codebook indices (see Table 4.2-1). After dropping the selected codebook indices, the remaining encoding parameters, which represent the new encoder output, are transmitted to the decoder along with a signal classification of I-HR. At the system interface with AMR-WB, the missing fixed-codebook indices are randomly generated and the frame is decoded as an AMR-WB frame at 12.65, 8.85, or 6.60 kbps, depending on the AMR-WB codec mode.

A special feature in case of maximum HR operation is that the thresholds used to distinguish between unvoiced and voiced frames are in general more relaxed to allow as many frames as possible to be encoded using the Unvoiced HR and Voiced HR encoding types. In case of maximum HR operation, in VMR-WB mode 0, the frame is declared an Unvoiced frame if local_VAD=0 or the following condition is satisfied

$$(\overline{R}_{xy3} < th_4) \quad \text{AND} \quad (e_t < th_5) \quad \text{AND} \quad (dE < th_6)$$

where $th_4 = 0.695$, $th_5 = 4$, $th_6 = 40$ (the thresholds used for VMR-WB mode 1 classification). The unvoiced classification in VMR-WB modes 1 and 2 is processed the same way as if the maximum rate would not be limited.

If maximum half-rate is enforced for the current frame, the Voiced HR encoding type selection is modified such that the normalized correlation threshold δ_g = 0.5 for all VMR-WB modes 0, 1 and 2 (Section 5.11.4).

If the maximum bit rate is limited to QR by the system and the signal is classified as unvoiced, then Unvoiced QR can be used. This is however possible only in the non-interoperable modes of VMR-WB (i.e., mode 0, 1, and 2), as the AMR-WB codec is unable to decode the QR frames. If Unvoiced QR cannot be used, the frame is encoded as if no maximum rate limitation was imposed and it is marked as erased frame for the bit packing. Similarly, if the imposed maximum rate is limited to ER, all active speech frames are encoded normally, but marked as erased. Then the frame is encoded as an Erasure ER frame or Erasure QR frame (using the lowest rate allowed) by means of an in-band signaling. This is done using the fact that the bits corresponding to ISF indices are not permitted to be all 0. This pattern is thus used for signaling an erasure. Further, to prevent generation of an all-zero packet by the VMR-WB encoder, the least significant bit of the last ISF byte is set to 1. This translates into setting the second data bit of ER frame to 0x01 and the 5^{th} data bit of QR frame to 0x01. All other bits are set to 0.

5.13 Quantization of the ISP Coefficients

Routine Name: `isf_enc`
Inputs:

- q_i : The immitance spectral pairs for current frame

Outputs:

- $\hat{q}_i$: The quantized immitance spectral pairs for current frame
- f_i : The immitance spectral frequencies for current frame

Initialization:

- The memory of the quantizer predictors are set to zero at initialization.

The LP filter coefficients are quantized using the ISP representation in the frequency domain; that is

$$f_i = \frac{fs}{2\pi}\arccos(q_i), \qquad i = 0,...14$$
$$= \frac{fs}{4\pi}\arccos(q_i), \qquad i = 15 \qquad (5.13\text{-}1)$$

where f_i are the ISFs in Hz [0,6400] and f_s = 12 800 is the sampling frequency. The ISF vector is given by $\mathbf{f}^t = [f_0, f_1, \ldots, f_{15}]$, with t denoting transpose. Either 1st order moving-average (MA) prediction or 1st order auto-regressive (AR) prediction is applied to the mean-removed ISF vector, and the prediction error vector is quantized using a combination of split vector quantization (SVQ) and multistage vector quantization (MSVQ). The prediction and quantization are performed as follows. Let $\mathbf{z}(n)$ denote the mean-removed ISF vector at frame n. The prediction residual vector $\mathbf{r}(n)$ is given by:

$$\mathbf{r}(n) = \mathbf{z}(n) - \mathbf{p}(n) \qquad (5.13\text{-}2)$$

where $\mathbf{p}(n)$ is the predicted ISF vector at frame n. The LP filter quantization will make use of the voice classification for the current frame. In all encoding types except Voiced HR, first order MA prediction is used where:

$$\mathbf{p}(n) = \alpha_{MA}\hat{\mathbf{r}}(n-1) \qquad (5.13\text{-}3)$$

where $\hat{\mathbf{r}}(n-1)$ is the quantized residual vector at the past frame and $\alpha_{MA} = 1/3$.

In order to improve the ISF quantization performance in case of Voiced HR frames, AR prediction is used. Voiced HR encoding type is used to encode stable voiced signals, whereby successive ISF vectors are strongly correlated. Thus the use of AR prediction results in a prediction error with lower dynamic range. Since the predictor is switched back to MA prediction for other types of frames and the spectrum is usually stable during Voiced HR frames, the effect of error propagation in case of frame erasures is well controlled. The predicted ISF vector in case of Voiced HR frames is given by

$$\mathbf{p}(n) = \alpha_{AR}\hat{\mathbf{z}}(n-1) \qquad (5.13\text{-}4)$$

where $\hat{\mathbf{z}}(n-1)$ is the mean-removed quantized ISF vector from previous frame and $\alpha_{AR} = 0.65$. When AR prediction is used, the prediction error in Equation (5.13-2) has a lower dynamic range. Thus, in order to use the same first stage quantization tables for the different rates and encoding types, the error vector $\mathbf{r}(n)$ is scaled to bring the dynamic range close to that of MA prediction. The scaling is given by

$$\mathbf{r}_\mathbf{s}(n) = S_q\mathbf{r}(n) \qquad (5.13\text{-}5)$$

where $S_q = 1.25$ in case of AR prediction and $S_q = 1$ in case of MA prediction (no scaling).

The ISF (scaled) prediction error vector $\mathbf{r}_\mathbf{s}$ is quantized using split-multistage vector quantization S-MSVQ. The vector is split into 2 subvectors $\mathbf{r}_1(n)$ and $\mathbf{r}_2(n)$ of dimensions 9 and 7, respectively. The 2 subvectors are quantized in two stages. In the first stage $\mathbf{r}_1(n)$ is quantized with 8 bits and $\mathbf{r}_2(n)$ with 8 bits. A schematic block diagram of ISF quantization using switched MA/AR prediction is shown in Figure 5.13-1. For FR and Unvoiced HR, the quantization error vectors $\mathbf{r}_i^{(2)} = \mathbf{r} - \hat{\mathbf{r}}_i, i = 1,2$ are split in the next stage into 3 and 2 subvectors, respectively. The subvectors are quantized using the bit-rates described in Table 5.13-1:

Table 5.13-1: Quantization of mean-removed ISF vector for the FR and Unvoiced HR.

1. Unquantized 16-element-long ISF vector				
2. Stage 1 (r_1) 8 bits			2. Stage 1 (r_2) 8 bits	
3. Stage 2 $\mathbf{r}_1^{(2)}$, 0-2 6 bits	3. Stage 2 $\mathbf{r}_1^{(2)}$, 3-5 7 bits	3. Stage 2 $\mathbf{r}_1^{(2)}$, 6-8 7 bits	3. Stage 2 $\mathbf{r}_2^{(2)}$, 0-2 5 bits	3. Stage 2 $\mathbf{r}_2^{(2)}$, 3-6 5 bits

For Generic HR and Voiced HR, the quantization error vectors $\mathbf{r}_i^{(2)} = \mathbf{r} - \hat{\mathbf{r}}_i$, *i*=1,2 are split in the next stage into 2 and 1 subvectors, respectively. The subvectors are quantized using the bit-rates described in Table 5.13-2.

Table 5.13-2: Quantization of ISF vector for Generic HR and Voiced HR

1. Unquantized 16-element-long ISF vector		
2. Stage 1 (r_1) 8 bits		2. Stage 1 (r_2) 8 bits
3. Stage 2 $\mathbf{r}_2^{(2)}$, 0-4 7 bits	3. Stage 2 $\mathbf{r}_1^{(2)}$, 5-8 7 bits	3. Stage 2 $\mathbf{r}_2^{(2)}$, 0-6 6 bits

For Unvoiced QR, the quantization error vectors $\mathbf{r}_i^{(2)} = \mathbf{r}_i - \hat{\mathbf{r}}_i$, *i*=1.2, are split in the next stage into 2 and 1 subvectors, respectively. The subvectors are quantized using the bit-rates described in Table 5.13-3.

Table 5.13-3: Quantization of ISF vector for Generic HR and Voiced HR

1. Unquantized 16-element-long ISF vector		
2. Stage 1 (r_1) 8 bits		2. Stage 1 (r_2) 8 bits
3. Stage 2 $\mathbf{r}_1^{(2)}$, 0-4 5 bits	3. Stage 2 $\mathbf{r}_1^{(2)}$, 5-8 5 bits	3. Stage 2 $\mathbf{r}_2^{(2)}$, 0-6 6 bits

The two 8-bit codebooks are shared by all ISF 46-bit, 36-bit, and 32-bit quantizers. However, in the case of the 36-bit quantizer used for Voiced HR frames, the 8-bit codebook of the 1st split has been slightly modified to suit better Voiced HR frame ISF quantization. Thus, 28 entries statistically less used by this coding type have been replaced by entries optimized for Voiced HR frames. A squared error distortion measure is used in the quantization process. In general, for an input ISF or error residual subvector $\mathbf{r}_i$, *i* = 1,2 and a quantized vector at index *k*, $\hat{\mathbf{r}}_i^k$, the quantization is performed by finding the index *k* which minimizes

$$E = \sum_{i=m}^{n} \left[r_i - \hat{r}_i^k \right]^2 \qquad (5.13\text{-}6)$$

where *m* and *n* are the indices of the first and last elements of the subvector.

Once the quantized (scaled) prediction error vector is found, inverse scaling is applied in case of AR prediction, that is

$$\hat{\mathbf{r}}(n) = \hat{\mathbf{r}}_s(n) / S_q, \qquad (5.13\text{-}7)$$

and mean-removed quantized ISF vector is given by

$$\hat{z}(n) = p(n) + \hat{\mathbf{r}}(n). \tag{5.13-8}$$

The quantized ISF vector is then found by adding the ISF-mean to the quantized mean-removed ISF vector. The memories of both MA and AR predictors are updated for use in next frame.

After quantization in the frequency domain, the ISF parameters are converted into the cosine domain to obtain the ISP vector $\hat{\mathbf{q}}$. Similar to the case of unquantized LP parameters, the quantized ISPs in the present and previous frames are interpolated to obtain a different quantized LP filter in every subframe (as in Section 5.6.5).

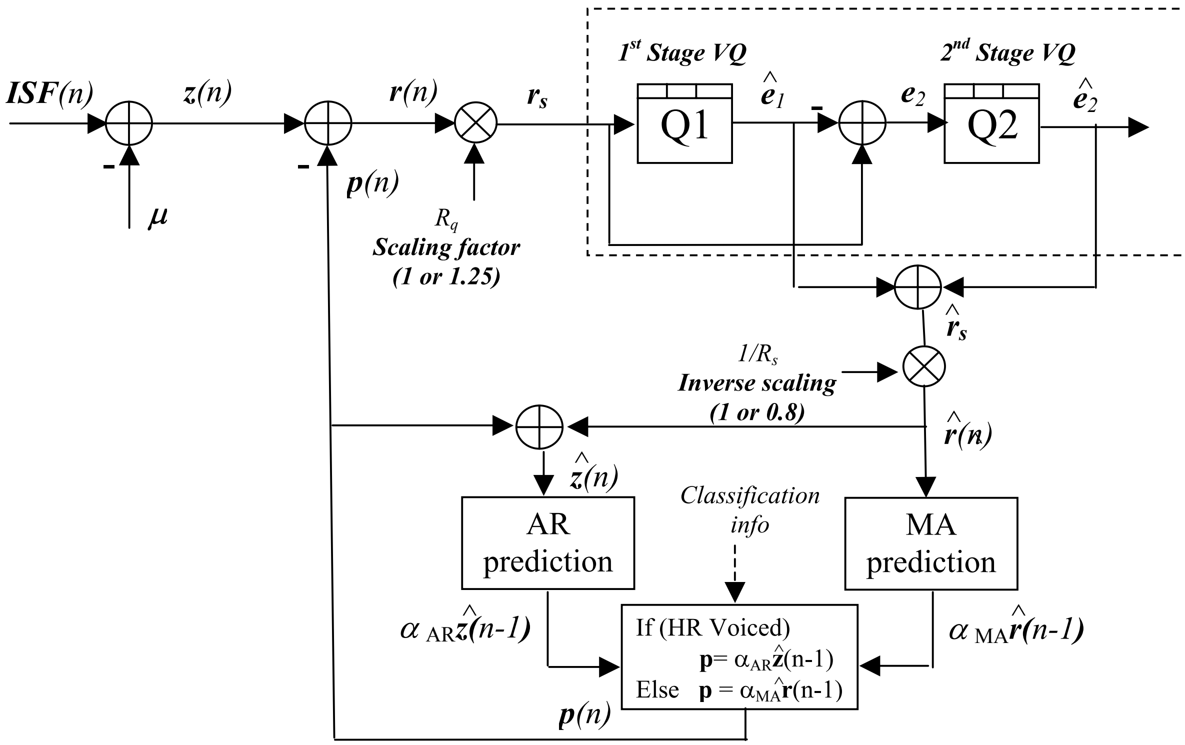

Figure 5.13-1: Block diagram of ISF quantization using switched MA/AR prediction.

5.14 Impulse Response Computation

__Routine Name:__ `vmr_encoder`

Inputs:

- a_i : The unquantized interpolated LP filter coefficients
- $\hat{a}_i$: The quantized interpolated LP filter coefficients

Outputs:

- $h(n)$: The impulse response of the synthesis filter

Initialization:

- None

The impulse response, $h(n)$, of the weighted synthesis filter

$$H(z)W(z) = A(z/\gamma_1)H_{\text{de-emph}}(z)/\hat{A}(z) \tag{5.14-1}$$

is computed for each subframe. The interpolation of the LP coefficients is described in Section 5.6.5. This impulse response is needed for the search of adaptive and fixed-codebooks. The impulse response $h(n)$ is computed by filtering the vector of coefficients of the filter $A(z/\gamma_1)$ extended by zeros through the two filters $1/\hat{A}(z)$ and $H_{\text{de-emph}}(z)$.

5.15 Target Signal Computation

Routine Name: `find_targets`
Inputs:

- *s*(*n*): speech signal
- a_i : The unquantized LP filter coefficients
- $\hat{a}_i$: The quantized LP filter coefficients

Outputs:

- *x*(*n*): The target signal for the adaptive codebook search
- *r*(*n*): The residual signal

Initialization:

- The filter memories are set to zero at initialization.

The target signal for adaptive codebook search is usually computed by subtracting the zero-input response of the weighted synthesis filter $H(z)W(z) = A(z/\gamma_1)H_{\text{de-emph}}(z)/\hat{A}(z)$ from the weighted speech signal $s_w(n)$. This is performed on a subframe basis.

An equivalent procedure for computing the target signal, which is used in this codec, is the filtering of the LP residual signal $r(n)$ through the combination of synthesis filter $1/\hat{A}(z)$ and the weighting filter $A(z/\gamma_1)H_{\text{de-emph}}(z)$. After determining the excitation for the subframe, the initial states of these filters are updated by filtering the difference between the LP residual and excitation. The memory update of these filters is explained in Section 5.21. The residual signal $r(n)$ which is needed for finding the target vector is also used in the adaptive codebook search to extend the past excitation buffer. This simplifies the adaptive codebook search procedure for delays less than the subframe size of 64 as will be explained in the next section. The LP residual is given by

$$r(n) = s(n) + \sum_{i=1}^{16} \hat{a}_i s(n-i), \qquad n = 0,\ldots,63 \tag{5.15-1}$$

5.16 Adaptive Codebook Search

Routine Name: `pit_encode, lp_filt_excitation_enc, gp_clip_test_isf, gp_clip, gp_clip_test_gain_pit`
Inputs:

- $r(n)$: The residual signal
- $u(n)$: Past excitation signal
- $x(n)$: The target signal for the adaptive codebook search
- $h(n)$: The impulse response of the weighted synthesis filter
- $\bar{d}_k$: The signal modification pitch delay parameter at the frame end boundary
- T_{op}: The open-loop estimates
- f_i : The immitance spectral frequencies for current frame

Outputs:

- $v(n)$: The adaptive codevector
- $y(n)$: The filtered adaptive codevector
- Integer and fractional close-loop pitch lag
- g_p: The adaptive codebook gain

Initialization:

- The buffers are set to zero at initialization. The 3-dimensional memory of the gain clipping procedure is initialized to 120, 0.6 and 0, respectively.

The adaptive codebook search is only performed in FR and Generic HR coding types. For Voiced HR encoding type, no closed-loop pitch search is performed since the pitch delay contour for the frame is determined as explained in Section 5.8.6. In this case, the adaptive codebook excitation is computed based in the delay contour. In case of Unvoiced and CNG-QR/ER encoding types, the adaptive codebook is not used.

The adaptive codebook search consists of performing closed-loop pitch search, and then computing the adaptive codevector, $v(n)$, by interpolating the past excitation at the selected fractional pitch lag. The adaptive codebook parameters (or pitch parameters) are the pitch delay and pitch gain g_p (adaptive codebook gain). In the search stage, the excitation is extended by the LP residual to simplify the closed-loop search.

5.16.1 Adaptive Codebook Search in Full Rate Encoding Type

For FR encoding types, adaptive codebook search is performed on a subframe basis. In the first and third subframes, a fractional pitch delay is used with resolutions ¼ in the range [34, $127\frac{3}{4}$], resolutions ½ in the range [128, $159\frac{1}{2}$], and integers only in the range [160, 231]. For the second and fourth subframes, a pitch resolution of 1/4 is always used in the range [T_p–8, T_p+7$\frac{3}{4}$], where T_p is nearest integer to the fractional pitch lag of the previous (1st or 3rd) subframe.

Closed-loop pitch analysis is performed around the open-loop pitch estimates on a subframe basis. In the first (and third) subframe the range $T_{op} \pm 7$, bounded by 34...231, is searched, where $T_{op} = d_0$ or d_1. For the other subframes, closed-loop pitch analysis is performed around the integer pitch selected in the previous subframe, as described above. In FR encoding types, the pitch delay is encoded with 9 bits in the first and third subframes and the relative delay of the other subframes is encoded with 6 bits. The closed-loop pitch search is performed by minimizing the mean-squared weighted error between the original and synthesized speech. This is achieved by maximizing

$$T_k = \frac{\sum_{n=0}^{63} x(n) y_k(n)}{\sqrt{\sum_{n=0}^{63} y_k(n) y_k(n)}} \qquad (5.16.1\text{-}1)$$

where $x(n)$ is the target signal and $y_k(n)$ is the past filtered excitation at delay k (past excitation convolved with $h(n)$). Note that the search range is limited around the open-loop pitch as explained earlier. The convolution $y_k(n)$ is computed for the first delay in the searched range, and for the other delays, it is updated using the recursive relation

$$y_k(n) = y_{k-1}(n-1) + u(-k)h(n) \qquad n=0,\ldots,63 \qquad (5.16.1\text{-}2)$$

where $u(n)$, n = –(231+17),…,63, is the excitation buffer. Note that in search stage, the samples $u(n)$, n = 0,...,63, are not known, and they are needed for pitch delays less than 64. To simplify the search, the LP residual is copied to $u(n)$ in order to make the relation in Equation (5.16.1-2) valid for all delays. Once the optimum integer pitch delay is determined, the fractions from $-\frac{3}{4}$ to $\frac{3}{4}$ with a step of $\frac{1}{4}$ around that integer are tested. The fractional pitch search is performed by interpolating the normalized correlation in Equation (5.16.1-1) and searching for its maximum. The interpolation is performed using an FIR filter for interpolating the term in Equation (5.16.1-1) using a Hamming windowed sinc function truncated at ± 17. The filter has its cut-off frequency (–3 dB) at 5050 Hz and –6 dB at 5760 Hz in the down-sampled domain, which means that the interpolation filter exhibit low-pass frequency response.

5.16.2 Adaptive Codebook Search in Generic HR

In the Generic HR encoding type, adaptive codebook search is performed twice per frame. That is, the adaptive codebook parameters are computed every half-frame. In the first half-frame, a fractional pitch delay is used with resolutions ½ in the range [34,91$\frac{1}{2}$], and integers only in the range [92, 231]. For the second half-frame a pitch resolution of ½ is always used in the range [T_1–8, T_1+7$\frac{1}{2}$], where T_1 is nearest integer to the fractional pitch lag of the first half-frame. In Generic HR, the adaptive codebook search is similar to FR with the difference that the excitation codevector size is 128 instead of 64.

5.16.3 Computation of Adaptive Codebook Excitation in FR and Generic HR

Once the fractional pitch lag is determined, the initial adaptive codebook excitation $v'(n)$ is computed by interpolating the past excitation signal $u(n)$ at the given phase (fraction). The interpolation is performed using an FIR filter (Hamming windowed sinc function) for interpolating the past excitation with the sinc truncated at ± 63. The filter has its cut-off frequency (–3 dB) at 5840 Hz and -6 dB at 6020 Hz in the down-sampled domain, which means that the interpolation filter exhibit low-pass frequency response. Thus, even when the pitch delay is an integer value, the adaptive codebook excitation consists of a low-pass filtered version of the past excitation at the given delay and not a direct copy thereof. Further, for delays smaller than the subframe size, the adaptive codebook excitation is completed based on the low-pass filtered interpolated past excitation and not by repeating the past excitation. In FR encoding types, the adaptive codebook excitation is computed for the subframe size of 64 samples. In the Generic HR, it is computed for 128 samples.

5.16.4 Computation of Adaptive Codebook Excitation in Voiced HR

In case of the Voiced HR encoding type, signal modification is used and a delay contour is computed for the whole frame as described in Equation (5.11.2-1). The initial adaptive codebook excitation $v'(n)$ in a certain subframe is computed by interpolating the past excitation in the adaptive codebook buffer at the delays given by the delay contour. The delay contour is computed using a 1/8 sub-sample resolution. The interpolation is performed using an FIR filter for interpolating the past excitation with the sinc truncated at ± 32. The filter has its cut-off frequency (–3 dB) at 5330 Hz and -6 dB at 6020 Hz in the down-sampled domain.

5.16.5 Frequency Dependent Pitch Prediction

In order to enhance the pitch prediction performance in wideband signals, a frequency-dependant pitch predictor is used. This is important in wideband signals since the periodicity does not necessarily extend over the whole spectrum. In this algorithm, there are two signal paths associated with respective sets of pitch codebook parameters, wherein each signal path comprises a pitch prediction error calculating device for calculating a pitch prediction error of a pitch codevector from a pitch codebook search mechanism. One of these two paths comprises a low-pass filter for filtering the pitch codevector before calculation of pitch prediction error. The pitch prediction error is then calculated for these two signal paths. The pitch prediction errors calculated for the two signal paths are compared, and the signal path having the lowest calculated pitch prediction error is selected, along with the associated pitch gain and pitch lag. The low-pass filter used in the second path is in the form $B_{LP}(z) = 0.18 + 0.64\,z^{-1} + 0.18z^{-2}$. Note that 1 bit is used to encode the chosen path.

Thus, there are two possibilities to generate the adaptive codebook $v(n)$, $v(n) = v'(n)$ in the first path, or $v(n) = \sum_{i=-1}^{1} b_{LP}(i+1)v'(n+i)$ in the second path, where $\mathbf{b}_{LP}$ = [0.18,0.64,0.18]. The path, which results in minimum energy of the target signal $x_2(n) = x(n) - g_p y(n),\ n = 0,...,63$, is selected for the filtered adaptive codebook vector, where g_p is the pitch gain computed as in the next section.

In case of FR encoding types, 1 bit per subframe is used indicate the use of low-pass filtering (4 bits per frame). In case of Voiced HR encoding type, the path filtering decision in the first subframe is extended to the second subframe. Similarly, the same low-pass filtering decision is used in both third and fourth subframes. Thus only 2 bits per frame are used for low-pass filtering in case of Voiced HR. In case of Generic FR, no low-pass filtering of the adaptive codebook excitation is used.

5.16.6 Computation of Adaptive Codebook Gain

The adaptive codebook gain, or the pitch gain, is then found by

$$g_p = \frac{\sum_{n=0}^{N-1} x(n)y(n)}{\sum_{n=0}^{N-1} y(n)y(n)}, \qquad \text{constrained by} \quad 0 \le g_p \le 1.2 \qquad (5.16.6\text{-}1)$$

where $y(n) = v(n) * h(n)$ is the filtered adaptive codebook vector (zero-state response of $H(z)W(z)$ to $v(n)$), and N=64. Note that the adaptive codebook vector may have been low-pass filtered as described in the previous section.

To avoid instability in case of channel errors, the adaptive codebook gain, or pitch gain, g_p is bounded by 0.95 if the pitch gains of the previous subframes have been close to 1 and the LP filters of the previous subframes have been close to being unstable (highly resonant).

The instability elimination method tests two conditions, resonance condition using the LP spectral parameters (minimum distance between adjacent ISFs), and gain condition by testing for high-valued pitch gains in the pervious frames. The method works as follows. First, the minimum distance between adjacent ISFs is computed as

$$d_{\min} = \min(ISF(i) - ISF(i-1)), \quad i = 1,...,14, \tag{5.16.6-2}$$

where the ISF frequencies are in the range [0,6400], then mean minimum distance is computed as

$$\bar{d}_{\min} = 0.8\bar{d}_{\min} + 0.2d_{\min} \quad \text{constrained by } \bar{d}_{\min} \le 120. \tag{5.16.6-3}$$

Second, the mean pitch gain is computed as

$$\bar{g} = 0.9\bar{g} + 0.1g_p \quad \text{constrained by } \bar{g} \ge 0.6. \tag{5.16.6-4}$$

If $\bar{g} \ge 0.9$ and $\bar{d}_{\min} \le 60$ then pitch gain clipping is performed by limiting pitch gain to $g_p = 0.95$. These conditions correspond to an average pitch gain of more that 0.9 and an average minimum distance of less than 60 Hz in the last 8 to 9 subframes approximately. For such signals, the instability at the output may happen in case of channel errors due to mismatch between the decoder and the encoder. Limiting the pitch gain to 0.9 in such conditions avoids this problem.

The initial values of $\bar{g}$ and $\bar{d}_{\min}$ are 0.6 and 120, respectively. Every time ER, QR, or Unvoiced HR are used, $\bar{g}$ and $\bar{d}_{\min}$ are reset to their initial values. In addition to the above mentioned methods, the adaptive codebook gain g_p is bounded by 0.95 also if the subframe energy of the target signal $x(n)$ drops by more than 6 dB relatively to the previous subframe and the mean pitch gain $\bar{g}$ > 1 at the same time. This measure prevents a potential divergence between floating-point and fixed-point implementations of the VMR-WB decoder.

5.17 Algebraic Codebook for FR, Voiced HR, and Generic HR

Routine Name: `inov_encode, find_targets`
Inputs:

- $r(n)$: The residual signal
- $x(n)$: The target signal for the adaptive codebook search
- $v(n)$: The adaptive codevector
- $y(n)$: The filtered adaptive codevector
- $h(n)$: The impulse response of the weighted synthesis filter
- Integer and fractional closed-loop pitch lag or the signal modification pitch delay parameter at the frame boundary
- g_p: The adaptive codebook gain

Outputs:

- $c(n)$: The algebraic codevector
- $z(n)$: The filtered algebraic codevector

Initialization:

- None

5.17.1 Codebook Structure

The codebook structure is based on interleaved single-pulse permutation (ISPP) design. The 64 positions in the codevector are divided into 4 tracks of interleaved positions, with 16 positions in each track. The different codebooks at the different rates are constructed by placing a certain number of signed pulses in the tracks (from 1 to 6 pulses per track). The codebook index, or codeword, represents the pulse positions and signs in each track. Thus, no codebook storage is needed, since the excitation vector at the decoder can be constructed through the information contained in the index itself (no lookup tables).

An important feature of this codebook is that it is a dynamic codebook, whereby the algebraic codevectors are filtered through an adaptive pre-filter $F(z)$. The transfer function of the adaptive pre-filter varies in time in relation to parameters representative of spectral characteristics of the signal to shape frequency characteristics of the excitation signal to damp frequencies perceptually annoying to the human ear. Here, a pre-filter relevant to wideband signals is used whereby $F(z)$ consists of two parts: a periodicity enhancement part $1/(1 - 0.85z^{-T})$ and a tilt part $(1 - \beta_1 z^{-1})$. That is,

$$F(z) = \frac{1-\beta_1 z^{-1}}{1-0.85z^{-T}} \qquad (5.17.1\text{-}1)$$

The periodicity enhancement part of the filter colors the spectrum by damping inter-harmonic frequencies, which are annoying to the human ear in case of voiced signals. In case of FR and Generic HR coding types, T is the integer part of the pitch lag (representing the fine spectral structure of the speech signal). In case of Voiced HR, T is computed on a sample basis following the pitch contour.

The factor β_1 of the tilt part of the pre-filter is related to the voicing of the previous subframe and is bounded by [0.0,0.5]. It is computed as

$$\beta_1 = \frac{0.5E'_v}{E'_v + E'_c} \qquad (5.17.1\text{-}2)$$

where E'_v and E'_c are the energies of the scaled pitch codevector and scaled innovation codevector of the previous subframe, respectively. The role of the tilt part is to reduce the excitation energy at low frequencies in case of voiced frames.

The codebook search is performed in the algebraic domain by combining the filter $F(z)$ with the weighed synthesis filter prior to the codebook search. Thus, the impulse response $h(n)$ must be modified to include the pre-filter $F(z)$. That is, $h(n) \leftarrow h(n) * f(n)$. The codebook structures of different encoding types are given below.

5.17.1.1 FR Encoding Types

In this codebook, the innovation vector contains 8 non-zero pulses. All pulses can have the amplitudes +1 or –1. The 64 positions in a subframe are divided into 4 tracks, where each track contains two pulses, as shown in Table 5.17-1.

Table 5.17-1: Potential positions of individual pulses in the algebraic codebook in FR

Track	Pulse	Positions
1	i_0, i_4	0, 4, 8, 12, 16, 20, 24, 28, 32 36, 40, 44, 48, 52, 56, 60
2	i_1, i_5	1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57, 61
3	i_2, i_6	2, 6, 10, 14, 18, 22, 26, 30, 34, 38, 42, 46, 50, 54, 58, 62
4	i_3, i_7	3, 7, 11, 15, 19, 23, 27, 31, 35, 39, 43, 47, 51, 55, 59, 63

Each two-pulse position in one track is encoded with 8 bits (total of 32 bits, 4 bits for the position of every pulse), and the sign of the first pulse in the track is encoded with 1 bit (total of 4 bits). This gives a total of 36 bits for the algebraic code.

In the case of two pulses per track of $K = 2^M$ potential positions (here M = 4), each pulse needs 1 bit for the sign and M bits for the position, which gives a total of $2M + 2$ bits. However, some redundancy exists due to the unimportance of the pulse ordering. For example, placing the first pulse at position p and the second pulse at position q is equivalent to placing the first pulse at position q and the second pulse at position p. One bit can be saved by encoding only one sign and deducing the second sign from the ordering of the positions in the index. Here the index is given by

$$I_{2p} = p_1 + p_0 \times 2^M + s \times 2^{2M} \qquad (5.17.1.1\text{-}1)$$

where s is the sign index of the pulse at position index p_0. If the two signs are equal then the smaller position is set to p_0 and the larger position is set to p_1. On the other hand, if the two signs are not equal, then the larger position is set to p_0 and the smaller position is set to p_1. At the decoder, the sign of the pulse at position p_0 is readily available. The second sign is deduced from the pulse ordering. If p_0 is larger than p_1 then the sign of the pulse at position p_1 is opposite to that at position p_0. If this is not the case, then the two signs are set equal

5.17.1.2 Voiced HR and Generic HR Encoding Types

In this codebook, the innovation vector contains 2 non-zero pulses. All pulses can have the amplitudes +1 or –1. The 64 positions in a subframe are divided into 2 tracks, where each track contains one pulse, as shown in Table 5.17-2.

Table 5.17-2: Potential positions of individual pulses in the algebraic codebook for Voiced HR and Generic HR

Track	Pulse	Positions
1	i_0	0, 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 62
2	i_1	1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63

Each pulse position in one track is encoded with 5 bits and the sign of the pulse in the track is encoded with 1 bit. This gives a total of 12 bits for the algebraic code.

The position index is given by the pulse position in the subframe divided by the pulse spacing (integer division). The division remainder gives the track index. For example, a pulse at position 31 has a position index of 31/2 = 15 and it belongs to the track with index 1 (second track). The sign index here is set to 0 for positive signs and 1 for negative signs. The index of the signed pulse is given by

$$\mathrm{I}_{1\mathrm{p}} = p + s \times 2^{M} \tag{5.17.1.2-1}$$

where p is the position index, s is the sign index, and M = 5 is the number of bits per track.

5.17.2 Algebraic Codebook Search

The algebraic codebook is searched by minimizing the mean square error between the weighted input speech and the weighted synthesis speech. The target signal used in the closed-loop pitch search is updated by subtracting the adaptive codebook contribution. That is

$$x_2(n) = x(n) - g_p y(n), \qquad n = 0,\dots,63 \tag{5.17.2-1}$$

where $y(n) = v(n)*h(n)$ is the filtered adaptive codebook vector and g_p is the unquantized adaptive codebook gain.

The matrix $\mathbf{H}$ is defined as the lower triangular Toeplitz convolution matrix with diagonal $h(0)$ and lower diagonals $h(1),\dots,h(63)$, and $\mathbf{d} = \mathbf{H}^t \mathbf{x}_2$ is the correlation between the target signal $x_2(n)$ and the impulse response $h(n)$ (also known as the backward filtered target vector), and $\Phi = \mathbf{H}^t\mathbf{H}$ is the matrix of correlations of $h(n)$. Here, $h(n)$ is the impulses response of the combination of the synthesis filter, the weighting filter, and the pre-filter $F(z)$ which includes a long-term filter.

The elements of the vector **d** are computed by

$$d(n) = \sum_{i=n}^{63} x_2(i) h(i-n), \qquad n = 0,\dots,63 \tag{5.17.2-2}$$

and the elements of the symmetric matrix Φ are computed by

$$\phi(i,j) = \sum_{n=j}^{63} h(n-i) h(n-j), \qquad i = 0,\dots,63 \qquad j = i,\dots,63 \tag{5.17.2-3}$$

If $\mathbf{c}_k$ is the algebraic codevector at index k, then the algebraic codebook is searched by maximizing the search criterion

$$Q_k = \frac{(\mathbf{x}_2^t \mathbf{H} \mathbf{c}_k)^2}{c_k^t \mathbf{H}^t \mathbf{H} c_k} = \frac{(\mathbf{d}^t \mathbf{c}_k)^2}{\mathbf{c}_k^t \mathbf{\Phi} c_k} = \frac{(R_k)^2}{E_k} \tag{5.17.2-4}$$

The vector **d** and the matrix Φ are usually computed prior to the codebook search.

The algebraic structure of the codebooks allows for very fast search procedures since the innovation vector $\mathbf{c}_k$ contains only a few nonzero pulses. The correlation in the numerator of Equation (5.17.2-4) is given by

$$R = \sum_{i=0}^{N_p - 1} s_i d(m_i) \tag{5.17.2-5}$$

where m_i is the position of the ith pulse, s_i is its amplitude (sign), and N_p is the number of pulses. The energy in the denominator of Equation (5.17.2-4) is given by

$$\mathrm{E} = \sum_{i=0}^{N_p-1} \phi(m_i, m_i) + 2\sum_{i=0}^{N_p-2} \sum_{j=i+1}^{N_p-1} s_i s_j \phi(m_i, m_j) \tag{5.17.2-6}$$

5.17.2.1 Codebook Search in FR Encoding Types

As explained above, a 36-bit algebraic codebook is used in Full-Rate, thereby efficient non-exhaustive search procedures are used to simplify the search. These procedures consist of signal-selected pulse amplitude for pre-setting the signs and depth-first tree search for determining the pulse positions. These procedures will be described below.

To simplify the search procedure, the pulse amplitudes are predetermined based on a certain reference signal $b(n)$. In this signal-selected pulse amplitude approach, the sign of a pulse at position i is set equal to the sign of the reference signal at that position. Here, the reference signal $b(n)$ is given by

$$b(n) = \sqrt{\frac{E_d}{E_r}}\, r_{LTP}(n) + \alpha d(n) \tag{5.17.2.1-1}$$

where $E_d = \mathbf{d}^t\mathbf{d}$ is the energy of the signal $d(n)$ and $E_r = \mathbf{r}_{LTP}^t \mathbf{r}_{LTP}$ is the energy of the signal $r_{LTP}(n)$ which is the residual signal after long-term prediction, and the entire amplitude information is given in terms of the sign of a pulse at position i being either +1 or –1. The scaling factor α controls the amount of dependence of the reference signal on $d(n)$, and it is lowered as the bit rate is increased. Here $\alpha = 1$ is used in FR encoding types.

To simplify the search the signal $d(n)$ and matrix Φ are modified to incorporate the pre-selected signs. Let $s_b(n)$ denote the vector containing the signs of $b(n)$. The modified signal $d'(n)$ is given by

$$d'(n) = s_b(n) d(n) \qquad n = 0,\ldots,N-1 \tag{5.17.2.1-2}$$

and the modified autocorrelation matrix Φ' is given by

$$\phi'(i,j) = s_b(i) s_b(j) \phi(i,j), \qquad i = 0,\ldots,N-1; \qquad j = i,\ldots,N-1 \tag{5.17.2.1-3}$$

The correlation at the numerator of the search criterion Q_k is now given by

$$R = \sum_{i=0}^{N_p-1} d'(m_i) \tag{5.17.2.1-4}$$

and the energy at the denominator of the search criterion Q_k is given by

$$\mathrm{E} = \sum_{i=0}^{N_p-1} \phi'(m_i, m_i) + 2\sum_{i=0}^{N_p-2} \sum_{j=i+1}^{N_p-1} \phi'(m_i, m_j) \tag{5.17.2.1-5}$$

Once the amplitudes have been pre-selected and incorporated into $d'(n)$ and $\phi'(i,j)$, the search will be confined to a subset of position/amplitude combinations by searching only the pulse amplitude/position combinations having non-zero-amplitude pulses with respect to the pre-selected amplitudes. Therefore, only those codewords having pulse positions in which the nonzero pulses agree in sign with the corresponding positions in $b(n)$ are used in computing R and E to obtain the search criterion $Q = (R)^2 / E$.

Thus, the goal of the search now is to determine the codevector with the best set of N_p pulse positions assuming amplitudes of the pulses have been selected as described above. The basic selection criterion is the maximization of the above-mentioned ratio Q_k.

In order to reduce the search complexity, a fast search procedure known as depth-first tree search procedure is used, whereby the pulse positions are determined N_m pulses at a time. More precisely, the N_p available pulses are partitioned into M non-empty subsets of N_m pulses respectively such that N_1+N_2...+N_m...+N_M = N_p. A particular choice of positions for the first J = N_1+N_2...+N_{m-1} pulses considered is called a level-m path or a path of length J. The basic criterion for a path of J pulse positions is the ratio $Q_k(J)$ when only the J relevant pulses are considered.

The search begins with subset #1 and proceeds with subsequent subsets according to a tree structure whereby subset m is searched at the mth level of the tree. The purpose of the search at level 1 is to consider the N_1 pulses of subset #1 and their valid positions in order to determine one, or a number of, candidate path(s) of length N_1, which are the tree nodes at level I. The path at each terminating node of level m–1 is extended to length N_1+N_2...+N_m at level m by considering N_m new pulses and their valid positions. One, or a number of, candidate extended path(s) are determined to constitute level-m nodes. The best codevector corresponds to that path of length N_p which maximizes the criterion $Q_k(N_p)$ with respect to all level-M nodes.

A special form of the depth-first tree search procedure is used here, in which two pulses are searched at a time, that is, $N_m = 2$, and these 2 pulses belong to two consecutive tracks. Further, instead of assuming that the matrix Φ is pre-computed and stored, which requires a memory of $N \times N$ words (64×64 = 4k words); a memory-efficient approach is used which reduces the memory requirement. In this approach, the search procedure is performed in such a way that only a part of the needed elements of the correlation matrix are pre-computed and stored. This part corresponds to the correlations of the impulse response corresponding to potential pulse positions in consecutive tracks, as well as the correlations corresponding to $\phi(j,j)$, $j = 0,\ldots,N-1$ (that is the elements of the main diagonal of matrix Φ).

In order to reduce the complexity, while testing possible combinations of two pulses, a limited number of potential positions of the first pulse are tested. Further, in case of a large number of pulses, some pulses in the higher levels of the search tree are fixed. In order to guess intelligently which potential pulse positions are considered for the first pulse, or in order to fix some pulse positions, a "pulse-position likelihood-estimate vector" **b** is used, which is based on speech-related signals. The pth component $b(p)$ of this estimate vector **b** characterizes the probability of a pulse occupying position p (p = 0, 1, ... N–1) in the best codevector that is being searched. Here the estimate vector **b** is the same vector used for pre-selecting the amplitudes and given in Equation (5.17.2.1-1).

The search procedures for all bit rate modes are similar. Two pulses are searched at a time, and these two pulses always correspond to consecutive tracks. That is the two searched pulses are in tracks T_0–T_1, T_1–T_2, T_2–T_3, or T_3–T_0. Before searching the positions, the sign of a pulse at potential position n is set to the sign of $b(n)$ at that position. Then the modified signal $d'(n)$ is computed as described above by including the predetermined signs.

For the first 2 pulses (1st tree level), the correlation at the numerator of the search criterion is given by

$$R = d'(m_0) + d'(m_1) \qquad (5.17.2.1\text{-}6)$$

and the energy at the denominator of the search criterion Q_k is given by

$$E = \phi'(m_0, m_0) + \phi'(m_1, m_1) + 2\phi'(m_0, m_1) \quad (5.17.2.1\text{-}7)$$

where the correlations $\phi'(m_i, m_j)$ has been modified to include the pre-selected signs at positions m_i and m_j.

For subsequent levels, the numerator and denominator are updated by adding the contribution of two new pulses. Assuming that two new pulses at a certain tree level with positions m_k and m_{k+1} from two consecutive tracks are searched, then the updated value of R is given by

$$R = R + d'(m_k) + d'(m_{k+1}) \quad (5.17.2.1\text{-}8)$$

and the updated energy is given by

$$E = E + \phi'(m_k, m_k) + \phi'(m_{k+1}, m_{k+1}) + 2\phi'(m_k, m_{k+1}) + 2R_{hv}(m_k) + 2R_{hv}(m_{k+1})$$

(5.17.2.1-9)

where $R_{hv}(m)$ is the correlation between the impulse response $h(n)$ and a vector $v_h(n)$ containing the addition of delayed versions of impulse response at the previously determined positions. That is,

$$v_h(n) = \sum_{i=0}^{k-1} h(n - m_i) \quad (5.17.2.1\text{-}10)$$

and

$$R_{hv}(m) = \sum_{n=m}^{N-1} h(n) v_h(n - m) \quad (5.17.2.1\text{-}11)$$

At each tree level, the values of $R_{hv}(m)$ are computed online for all possible positions in each of the two tracks being tested. It can be seen from Equation (5.17.2.1-9) that only the correlations $\phi'(m_k, m_{k+1})$ corresponding to pulse positions in two consecutive tracks need to be stored ($4 \times 16 \times 16$ words), along with the correlations $\phi'(m_k, m_k)$ corresponding to the diagonal of the matrix Φ (64 words). Thus the memory requirement in the present algebraic structure is 1088 words instead of $64 \times 64 = 4096$ words.

In the FR encoding types, 2 pulses are placed in each track giving a total of 8 pulses per subframe of length 64. Two pulses are searched at a time, and these two pulses always correspond to consecutive tracks. That is the two searched pulses are in tracks T_0–T_1, T_1–T_2, T_2–T_3, or T_3–T_0. The tree has 4 levels in this case. At the first level, pulse P_0 is assigned to track T_0 and pulse P_1 to track T_1. In this level, no search is performed and the two pulse positions are set to the maximum of $b(n)$ in each track. In the second level, pulse P_2 is assigned to track T_2 and pulse P_3 to track T_3. 4 positions for pulse P_2 are tested against all 16 positions of pulse P_3. The 4 tested positions of P_2 are determined based on the maxima of $b(n)$ in the track. In the third level, pulse P_4 is assigned to track T_1 and pulse P_5 to track T_2. 8 positions for pulse P_4 are tested against all 16 positions of pulse P_5. Similar to the previous search level, the 8 tested positions of P_4 are determined based on the maxima of $b(n)$ in the track. In the fourth level, pulse P_6 is assigned to track T_3 and pulse P_7 to track T_0. 8 positions for pulse P_6 are tested against all 16 positions of pulse P_7. Thus the total number of tested combinations is $4 \times 16 + 8 \times 16 + 8 \times 16 = 320$. The whole process is repeated 4 times (4 iterations) by assigning the pulses to different tracks. For example, in the 2nd iteration, pulses P_0 to P_7 are assigned to tracks T_1, T_2, T_3, T_0, T_2, T_3, T_0, and T_1, respectively. Thus the total number of tested position combinations is $4 \times 320 = 1280$.

Once the pulse positions and signs are determined, the algebraic codevector is constructed then the fixed-codebook excitation vector is found by filtering the algebraic codevector through the pre-filter $F(z)$.

5.17.2.2 Codebook Search in Voiced HR and Generic HR

In the Voiced HR and Generic HR encoding types, as explained above a 12 bit codebook is used. Due to the reasonable codebook size, exhaustive search is used. Since there are only two pulses, the computation of the correlation and energy terms in Equation (5.17.2-4) is simple. The correlation in the numerator of Equation (5.17.2-6) is given by

$$R = s_0 d(m_0) + s_1 d(m_1) \tag{5.17.2.2-1}$$

and the energy at the denominator of the search criterion Q_k is given by

$$E = \phi(m_0, m_0) + \phi(m_1, m_1) + 2 s_0 s_1 \phi(m_0, m_1) \tag{5.17.2.2-2}$$

The search criterion is computed for all possible position and sign combinations to find the optimum pulse positions and signs. The correlations of the impulse response corresponding to potential pulse positions in the two tracks are only computed and stored, as well as the correlations corresponding to $\phi(j,j)$, $j = 0,\ldots,63$ (that is the elements of the main diagonal of matrix Φ). This gives a total of $32\times32+64=1088$ as in the FR case.

Once the pulse positions and signs are determined, the algebraic codevector is constructed then the fixed-codebook excitation vector is found by filtering the algebraic codevector through the pre-filter $F(z)$.

5.18 Gaussian Codebook Structure and Search in Unvoiced HR

<u>Routine Name</u>: `gaus_encode`
Inputs:

- $x(n)$: The target signal
- $h(n)$: The impulse response of the weighted synthesis filter

Outputs:

- $c(n)$: The Gaussian codevector
- $z(n)$: The filtered Gaussian codevector
- g_c: The Gaussian codebook gain

Initialization:

- None

5.18.1 Structure of the random codebook

In Unvoiced HR, a Gaussian codebook is used for representing the excitation. To simplify the search and reduce the codebook memory requirement, an efficient structure is used whereby the excitation codevector is derived by the addition of 2 signed vectors taken from a table containing 64 random vectors of dimension N (here N is the subframe size 64). Let $\mathbf{v}_i$ denote the ith N-dimensional random vector in the random table, then a codevector is constructed by

$$\mathbf{c} = s_1 \mathbf{v}_{p_1} + s_2 \mathbf{v}_{p_2} \qquad (5.18.1\text{-}1)$$

where the signs s_1 and s_2 are signs equal to -1 or 1, and p_1 and p_2 are the indices of the random vectors from the random table. In order to reduce the table memory, a shift-by-2 table is used, thus only 64+63×2=190 values are needed to represent the 64 vectors of dimension N=64.

To encode the codebook index, one has to encode 2 signs, s_1 and s_2, and two indices, p_1 and p_2. The values of p_1 and p_2 are in the range 0 to 63, so they need 6 bits each, and the signs need 1 bit each. However, 1 bit can be saved since the order of the vectors $\mathbf{v}_i$ and $\mathbf{v}_j$ is not important. For example, choosing $\mathbf{v}_{16}$ as the first vector and $\mathbf{v}_{25}$ as the second vector is equivalent to choosing $\mathbf{v}_{25}$ as the first vector and $\mathbf{v}_{16}$ as the second vector. Thus, similar to the case of encoding two pulses in a track, only one bit can be used for both signs while ordering the vector indices in a way such that the other sign information can be easily deduced. This gives a total of 13 bits. To better explain this procedure, assume that the two vectors have the indices p_1 and p_2 with sign indices σ_1 and σ_2, respectively (σ=0 if the sign is positive and σ=1 if the sign is negative). The codevector index is given by

$$I = \sigma_1 + 2 \times (p_1 \times 6 + p_2) \qquad (5.18.1\text{-}2)$$

If $p_1 \le p_2$ then $\sigma_2 = \sigma_1$; otherwise σ_2 is different from σ_1. Thus, when constructing the codeword (index of codevector), if the two signs are equal then the smaller index is assigned to p_1 and the larger index to p_2, otherwise the larger index is assigned to p_1 and the smaller index to p_2.

5.18.2 Search of the Random Codebook

The goal of the search procedure is to find the indices p_1 and p_2 of the best 2 vectors and their corresponding signs s_1 and s_2, which maximize the search criterion

$$Q_k = \frac{(\mathbf{x}^t \mathbf{z}_k)^2}{\mathbf{z}_k^t \mathbf{z}_k} = \frac{(\mathbf{x}^t \mathbf{H} \mathbf{c}_k)^2}{\mathbf{z}_k^t \mathbf{z}_k} = \frac{(\mathbf{d}^t \mathbf{c}_k)^2}{\mathbf{z}_k^t \mathbf{z}_k} \qquad (5.18.1\text{-}3)$$

where $\mathbf{x}$ is the target vector and $\mathbf{z}_k = \mathbf{H}\mathbf{c}_k$ is the filtered codevector at index k. Note that in the numerator of the search criterion, the dot product between $\mathbf{x}$ and $\mathbf{z}_k$ is equivalent to the dot product between $\mathbf{d}$ and $\mathbf{c}_k$, where $\mathbf{d}=\mathbf{H}^t\mathbf{x}$ is the backward filtered target vector which is also the correlation between $\mathbf{d}$ and the impulse response $\mathbf{h}$. The elements of the vector $\mathbf{d}$ are found by

$$d(n) = x(n) * h(-n) = \sum_{i=n}^{N-1} x(i) h(i-n) \qquad (5.18.1\text{-}4)$$

Since $\mathbf{d}$ is independent of the codevector index k, it is computed only once, which simplifies the computation of the numerator for the different codevectors.

After computing the vector $\mathbf{d}$, a pre-determination process is used to identify K out of the 64 random vectors in the random table, so that the search process is then confined to those K vectors. The pre-determination is performed by testing the numerator of the search criterion Q_k for the K vectors which have the largest absolute dot product (or squared dot product) between $\mathbf{d}$ and $\mathbf{v}_i$, i=0,...,63. That is, the dot products χ_i given by

$$\chi_i = \sum_{n=0}^{N-1} d(n) v_i(n) \qquad (5.18.1\text{-}5)$$

are computed for all random vectors $\mathbf{v}_i$ and the indices of the K vectors which result in the K largest values of $|\chi_i|$ are retained. These indices are stored in the index vector m_i, i=0,...,K-1. To further simplify the search, the sign information corresponding to each pre-determined vector is also pre-set. The sign corresponding to each pre-determined vector is given by the sign of χ_i for that vector. These pre-set signs are stored in the sign vector s_i, i=0,...,K-1.

The codebook search is now confined to the pre-determined K vectors with their corresponding signs. Here, the value K=8 is used, thus the search is reduced to finding the best 2 vectors among 8 random vectors instead of finding them among 64 random vectors. This reduces the number of tested vector combinations from 64×65/2=2080 to 8×9/2=36.

Once the best promising K vectors and their corresponding signs are pre-determined, the search proceeds for selecting 2 vectors among those K vectors which maximize the search criterion Q_k.

We first start by computing and storing the filtered vectors $\mathbf{w}_j$, j=0,…,K-1 corresponding to the K pre-determined vectors. This can be performed by convolving the pre-determined vectors with the impulse response of the weighted synthesis filter $h(n)$. The sign information is also included in the filtered vectors; that is

$$w_j(n) = s_j \sum_{i=0}^{n} v_{m_j}(i)h(n-i), \qquad n=0,...,63, \qquad j=0,...,K-1. \tag{5.18.1-6}$$

We then compute the energy of each filtered pre-determined vector

$$\varepsilon_j = \mathbf{w}_j^t \mathbf{w}_j = \sum_{n=0}^{63} w_j^2(n), \qquad j=0,...,K-1 \tag{5.18.1-7}$$

and its dot product with the target vector

$$\rho_j = \mathbf{x}^t \mathbf{w}_j = \sum_{n=0}^{63} w_j(n)x(n), \qquad j=0,...,K-1. \tag{5.18.1-8}$$

Note that ρ_j and ε_j correspond to the numerator and denominator of the search criterion due to each predetermined vector. The search proceeds now with the selection of 2 vectors among the K pre-determined vectors by maximizing the search criterion Q_k. Note that the codevector is given by

$$\mathbf{c} = s_1 \mathbf{v}_{p_1} + s_2 \mathbf{v}_{p_2} \tag{5.18.1-9}$$

The filtered codevector **z** is given by

$$\mathbf{z} = \mathbf{Hc} = s_1 \mathbf{Hv}_{p_1} + s_2 \mathbf{Hv}_{p_2} = \mathbf{w}_{p_1} + \mathbf{w}_{p_2} \tag{5.18.1-10}$$

Note that the pre-determined signs are included in the filtered pre-determined vectors $\mathbf{w}_i$. The search criterion is given by (the codevector index k is dropped for simplicity)

$$Q = \frac{(\mathbf{x}^t\mathbf{z})^2}{\mathbf{z}^t\mathbf{z}} = \frac{(\mathbf{x}^t\mathbf{w}_{p_1} + \mathbf{x}^t\mathbf{w}_{p_2})^2}{(\mathbf{w}_{p_1} + \mathbf{w}_{p_2})^t(\mathbf{w}_{p_1} + \mathbf{w}_{p_2})} = \frac{(\rho_{p_1} + \rho_{p_2})^2}{\varepsilon_{p_1} + \varepsilon_{p_2} + 2\mathbf{w}_{p_1}^t\mathbf{w}_{p_2}} \tag{5.18.1-11}$$

The vectors $\mathbf{w}_j$ and the values of ρ_j and ε_j are computed before starting the codebook search. The search is performed in two nested loops for all possible positions p_1 and p_2 that maximize the search criterion Q. Only the dot products between the different vectors $\mathbf{w}_j$ need to be computed inside the loop.

At the end of the two nested loops, the optimum vector indices p_1 and p_2 will be known. The two indices and the corresponding signs are then encoded as described above. The gain of the excitation vector is computed based on a combination of waveform matching and energy matching. The gain is given by

$$g_c = 0.6 g_w + 0.4 g_e \qquad (5.18.1\text{-}12)$$

where g_w is the gain that matches the waveforms of the vectors **x** and **z** and given by $g_w = \frac{\mathbf{x}^t\mathbf{z}}{\mathbf{z}^t\mathbf{z}}$ and g_e is the gain that matches the energies of the vectors **x** and **z** and given by $g_e = \sqrt{\frac{\mathbf{x}^t\mathbf{x}}{\mathbf{z}^t\mathbf{z}}}$. Here, **x** is the target vector and **z** is the filtered selected excitation vector **c** given in Equation (5.18.1-1).

5.19 Random Excitation in Unvoiced QR

Routine Name: `gaus_encode`
Inputs:

- $x(n)$: The target signal
- $h(n)$: The impulse response of the weighted synthesis filter

Outputs:

- $c(n)$: The Gaussian codevector
- $z(n)$: The filtered Gaussian codevector
- g_c: The Gaussian codebook gain

Initialization:

- None

In Unvoiced QR encoding type, no bits are used to encode the excitation signal. The excitation signal is derived from the same random table used in the Unvoiced HR encoding type and in the same manner. That is, the excitation vector is given by

$$\mathbf{c} = s_1 \mathbf{v}_{p_1} + s_2 \mathbf{v}_{p_2}\,. \qquad (5.19\text{-}1)$$

However, the signs and indices of the two vectors are randomly generated. Since in Unvoiced HR a 13-bit random codebook is used, in Unvoiced QR, a 13-bit integer is randomly generated from which the indices and signs of the two vectors are derived.

To avoid a low-frequency artifact in case of misclassification of a voiced offset as an unvoiced frame, the following measure is applied to the randomly generated excitation vector ***c***. First the distance between the first two ISFs is computed. If this distance is less than 100 Hz, the tilt c_{tilt} of the excitation vector is estimated as

$$c_{tilt} = \sum_{i=1}^{N-1} c_{i-1} c_i$$

where N is the subframe length. If c_{tilt} > 0, the sign of every other sample of the excitation vector is reversed (c_{2i} is multiplied by –1) in order to prevent vector $\mathbf{c}$ of having a low-frequency characteristic.

Since no waveform matching is used to determine the excitation vector, the excitation gain is determined based only on matching the energies of the target vector **x** and the filtered excitation vector **z**=**c*****h**. The excitation gain is given by

$$g_c = \frac{1}{\sqrt{1.84}}\sqrt{\frac{\mathbf{x}^t\mathbf{x}}{\mathbf{z}^t\mathbf{z}}}. \tag{5.19-2}$$

The scaling factor is introduced for the following reason. In Unvoiced HR the gain is computed as a weighted sum of waveform matching and energy matching. Since in the Unvoiced QR case energy matching alone is performed, the energy of the Unvoiced QR portions of the signal were found to be higher than the case where Unvoiced HR encoding was used for the same portions. The value 1.84 has been found experimentally by comparing the energy of the Unvoiced QR portions with the energy of the same portions coded by Unvoiced HR, using a database consisting of nominal, low and high level signals.

5.20 Quantization of the Adaptive and Fixed-Codebook Gains

Routine Name: `gain_enc_2`

Inputs:

- $x(n)$: The target signal
- $y(n)$: The filtered adaptive codevector
- $c(n)$: The algebraic codevector
- $z(n)$: The filtered algebraic codevector
- g_p: The adaptive codebook gain

Outputs:

- $\hat{g}_p$: The quantized adaptive codebook gain
- $\hat{g}_c$: The quantized algebraic codebook gain

Initialization:

- The quantized energy prediction errors $\hat{R}(k)$ are set to –14.

In FR, Generic HR, and Voiced HR encoding types, the adaptive codebook gain (pitch gain) and the fixed (algebraic) codebook gain are vector quantized using a 128-element codebook. However, in Generic HR and Voiced HR, in every two subframes, either the lower or higher half of the codebook is used.

The fixed-codebook gain quantization is performed using MA prediction with fixed coefficients. The 4th order MA prediction is performed on the innovation energy as follows. Let $E(n)$ be the mean-removed innovation energy (in dB) at subframe n, and given by

$$E(n) = 10\log\left(\frac{1}{N}g_c^2\sum_{i=0}^{N-1}c^2(i)\right) - \overline{E} \tag{5.20-1}$$

where N = 64 is the subframe size, $c(i)$ is the fixed-codebook excitation, and $\overline{E} = 30$ dB is the mean of the innovation energy. Equation (5.20-1) can be expressed by

$$E(n) = E_i + G_c - \overline{E} \tag{5.20-2}$$

where

$$E_i = 10\log\left(\frac{1}{N}\sum_{i=0}^{N-1} c^2(i)\right) \tag{5.20-3}$$

is the mean innovation energy in dB and

$$G_c = 20\log(g_c) \tag{5.20-4}$$

is the innovation gain in dB. The predicted energy is given by

$$\widetilde{E}(n) = \sum_{i=1}^{4} b_i \hat{R}(n-i) \tag{5.20-5}$$

where [b_1 b_2 b_3 b_4] = [0.5,0.4,0.3,0.2] are the MA prediction coefficients, and $\hat{R}(k)$ is the quantized energy prediction error at subframe k. The predicted energy is used to compute a predicted fixed-codebook gain g'_c as in Equation (5.20-1) (by substituting $E(n)$ by $\widetilde{E}(n)$ and g_c by g'_c). This is done as follows. First, the mean innovation energy E_i is found as in Equation (5.20-3) and then the predicted gain G'_c in dB is found by

$$G'_c = \widetilde{E}(n) + \overline{E} - E_i \tag{5.20-6}$$

The prediction gain in the linear domain is given by

$$g'_c = 10^{0.05G_c} = 10^{0.05(\widetilde{E}(n)+\overline{E}-E_i)} \tag{5.20-7}$$

A correction factor between the gain g_c and the estimated one g'_c is given by

$$\gamma = g_c / g'_c \tag{5.20-8}$$

Note that the prediction error is given by

$$R(n) = E(n) - \widetilde{E}(n) = 20\log(\gamma) \tag{5.20-9}$$

The pitch gain, g_p, and correction factor γ are jointly vector quantized using a 7-bit codebook. The gain codebook search is performed by minimizing the mean-squared of the weighted error between original and reconstructed speech, which is given

$$E = \mathbf{x}^t\mathbf{x} + g_p^2\mathbf{y}^t\mathbf{y} + g_c^2\mathbf{z}^t\mathbf{z} - 2g_p\,\mathbf{x}^t\mathbf{y} - 2g_c\mathbf{x}^t\mathbf{z} + 2g_p\,g_c\mathbf{y}^t\mathbf{z} \tag{5.20-10}$$

where the x is the target vector, y is the filtered adaptive codebook vector, and z is the filtered fixed-codebook vector. (Each gain vector in the codebook also has an element representing the quantized energy prediction error.) The quantized energy prediction error associated with the chosen gains is used to update $\hat{R}(n)$.

In the FR encoding types, 7 bits are used in every subframe; however, in the search stage, only the 64 codevectors that are closest to the unquantized pitch gain, g_p, are taken into account. Since the two dimensional vectors in the codebook are ordered according to the pitch gain value, the initial index closest to the unquantized pitch gain is determined and the search is confined to the 64 vectors around that index.

In the Voiced HR and Generic HR cases, only half of the 7-bit codebook is used. Thus, 6 bits are needed in every subframe for encoding the index in the codebook half, but 1 bit is needed once every two subframes to indicate which codebook half is used. This gives a total of 6×4+2=26 bits for encoding the gains in the 4 subframes.

To decide which codebook half is used in the first two subframes, an initial pitch gain is computed based on two subframes as

$$g_i = \frac{\sum_{n=0}^{2N-1} x(n)y(n)}{\sum_{n=0}^{2N-1} y(n)y(n)} \qquad (5.20\text{-}11)$$

This is similar to Equation (5.16.6-1) but with the summation performed over two subframes. The computation of the target signal $x(n)$ and the filtered pitch codebook signal $y(n)$ is also performed over a period of two subframes. Computing the target signal $x(n)$ over a period longer than one subframe is performed by extending the computation of the weighted speech signal $s_w(n)$ and the zero input response $\boldsymbol{s_0}$ over a longer period while using the same LP filter in first subframe for all the extended period. The target signal $x(n)$ is computed as the weighted speech signal $s_w(n)$ after subtracting the zero-input response $\boldsymbol{s_0}$ of the weighted synthesis filter $W(z)/\hat{A}(z)$. Similarly, computation of the filtered adaptive codebook signal $y(n)$ is performed by extending the computation of the adaptive codebook vector $v(n)$ and the impulse response $h(n)$ of the weighted synthesis filter $W(z)/\hat{A}(z)$ of the first subframe over a period longer than the subframe length. The filtered adaptive codebook signal is the convolution between the adaptive codebook vector $v(n)$ and the impulse response $h(n)$, where the convolution in this case is computed over two subframes.

The pitch gain value at index 64 in the codebook is 0.768606. In the first subframe, if the initial pitch gain calculated over two subframes is larger than or equal to 0.768606 then the upper half of the codebook is used in the first two subframes; otherwise the lower half is used. The same procedure described above is used in the gain quantization of the third and fourth subframes.

Furthermore, if pitch gain clipping is detected (as described above), the last 27 entries in the codebook are skipped in the quantization procedure since the pitch gain in these entries is higher than 1.

5.20.1 Gain Quantization in Unvoiced HR and Unvoiced QR

In Unvoiced HR and QR encoding types, the adaptive codebook is not used and only the innovation gain needs to be quantized. Gain prediction is done in the same manner as described above. First, a predicted gain G'_c is computed as described in Equation (5.20-6). Then the gain prediction error in subframe n is computed as

$$\Gamma = R(n) = 20\log(g_c) - G'_c \qquad (5.20.1\text{-}1)$$

Note that this is equivalent to Equation (5.20-8) but in dB. The gain prediction error in dB is uniformly quantized between $\Gamma_{\min}$ and $\Gamma_{\max}$ with step size given by

$$\delta = (\Gamma_{\min} - \Gamma_{\max}) / L \tag{5.20.1-2}$$

where L is the number of quantization levels. The quantization index k is given by the integer part of

$$\frac{\Gamma - \Gamma_{\min}}{\delta} + 0.5 \tag{5.20.1-3}$$

and the quantized prediction error is given by

$$\hat{\Gamma} = \hat{R}(n) = k \times \delta + \Gamma_{\min} \tag{5.20.1-4}$$

Finally the quantized gain is given by

$$\hat{g}_c = 10^{0.05(\hat{\Gamma} + G'_c)} \tag{5.20.1-5}$$

In Unvoiced HR, 6 bits are used to quantize the gain, thus L_v =64, and the quantization boundaries are $\Gamma_{\min} = -30$ and $\Gamma_{\max} = 34$ (the quantization step is 1 dB).

In Unvoiced QR, 5 bits are used to quantized the gain, thus L_v =32, and the quantization boundaries are $\Gamma_{\min} = -22$ and $\Gamma_{\max} = 20$ (the quantization step is also 1.3125 dB).

5.21 Memory Update

An update of the states of the synthesis and weighting filters is needed in order to compute the target signal in the next subframe.
After the two gains have been quantized, the excitation signal, $u(n)$, in the present subframe is found by

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c(n), \qquad n = 0,...,63 \tag{5.21-1}$$

where $\hat{g}_p$ and $\hat{g}_c$ are the quantized adaptive and fixed-codebook gains, respectively, $v_i(n)$ the adaptive codebook vector (interpolated past excitation), and $c(n)$ is the fixed-codebook vector (algebraic code including pre-filtering). The states of the filters can be updated by filtering the signal $r(n)$–$u(n)$ (difference between residual and excitation) through the filters $1/\hat{A}(z)$ and $A(z/\gamma_1)H_{\text{de-emph}}(z)$ for the 64 sample subframe and saving the states of the filters. This would require 3 stages of filtering. A simpler approach, which requires only one filtering, is as follows. The local synthesis speech, $\hat{s}(n)$, is computed by filtering the excitation signal through $1/\hat{A}(z)$. The output of the filter due to the input $r(n)$–$u(n)$ is equivalent to $e(n) = s(n) - \hat{s}(n)$. So the states of the synthesis filter $1/\hat{A}(z)$ are given by $e(n)$, n = 48,…,63. Updating the states of the filter $A(z/\gamma_1)H_{\text{de-emph}}(z)$ can be done by filtering the error signal $e(n)$ through this filter to find the perceptually weighted error $e_w(n)$. However, the signal $e_w(n)$ can be equivalently found by

$$e_w(n) = x(n) - \hat{g}_p y(n) - \hat{g}_c z(n) \tag{5.21-2}$$

Since the signals $x(n)$, $y(n)$, and $z(n)$ are available, the states of the weighting filter are updated by computing $e_w(n)$ as in Equation (5.21-2) for n = 48,...,63. This saves two stages of filtering.

5.22 Supplementary Information for Frame Error Concealment in Generic FR

Routine Name: `fer_encode`
Inputs:

- $s(n)$: The speech signal
- $\hat{s}(n)$: The synthesized speech signal
- $s_w(n)$: The weighted speech signal
- $\hat{s}_w(n)$: The weighted synthesis signal
- Encoding type
- $\bar{d}_k$: The signal modification pitch delay parameter at the frame boundary
- d_0, d_1, and d_2 : The pitch lags in each half-frame
- $C_{norm}(d)$: The normalized correlation at pitch lags d_1 and d_2
- The local VAD flag
- $e_{tilt}(i)$: Spectral tilt
- The closed-loop pitch lags
- E_{rel}: Relative frame energy
- Classification decision of previous frame

Outputs:

- Frame classification decision
- E: The synthesized speech energy
- T_0: First glottal pulse position with respect to frame beginning

Initialization:
- None

In the Generic FR encoding type, 14 bits are used to send supplementary information, which improves frame erasure concealment and the convergence and recovery of the decoder after erased frames. These parameters include energy information (6 bits), signal classification information (2 bits), and phase information (the estimated position of the first glottal pulse in a frame) (6 bits). In the next sections, computation and quantization of these additional parameters will be described in detail.

5.22.1 Signal Classification for Frame Error Concealment and Recovery

The basic idea behind using a classification of the speech for a signal reconstruction in the presence of erased frames consists of the fact that the ideal concealment strategy is different for quasi-stationary speech segments and for speech segments with rapidly changing characteristics. While the best processing of erased frames in non-stationary speech segments can be summarized as a rapid convergence of speech-encoding parameters to the ambient noise characteristics, in the case of quasi-stationary signal, the speech-encoding parameters do not vary significantly and can be kept practically unchanged during several adjacent erased frames before being damped. Also, the optimal method for a signal recovery following an erased block of frames varies with the classification of the speech signal.

The speech signal can be roughly classified as voiced, unvoiced and pauses. Voiced speech contains an important amount of periodic components and can be further divided in the following categories: voiced onsets, voiced segments, voiced transitions and voiced offsets. A voiced onset is defined as a beginning of a voiced speech segment after a pause or an unvoiced segment. During voiced segments, the speech signal parameters (spectral envelope, pitch period, ratio of periodic and non-periodic components, energy) vary slowly from frame to frame. A voiced transition is characterized by rapid variations of a voiced speech, such as a transition between vowels. Voiced offsets are characterized by a gradual decrease of energy and voicing at the end of voiced segments.
The unvoiced parts of the signal are characterized by missing the periodic component and can be further divided into unstable frames, where the energy and the spectrum changes rapidly, and stable frames where these characteristics remain relatively stable. Remaining frames are classified as silence. Silence frames comprise all frames without active speech, i.e. also noise-only frames if a background noise is present.

Not all of the above mentioned classes need a separate processing. Hence, for the purposes of error concealment techniques, some of the signal classes are grouped together.

In determining the supplementary parameters, the lookahead at the encoder is used. The lookahead allows estimate of the evolution of the signal in the following frame and consequently the classification can be done by taking into account the future signal behavior.

The frame classification is done with the consideration of the concealment and recovery strategy. In other words, any frame is classified in such a way that the concealment can be optimal if the following frame is missing, or that the recovery can be optimal if the previous frame was lost. Some of the classes used for the FER processing need not be transmitted, as they can be deduced without ambiguity at the decoder. Five distinct classes are used, and defined as follows:

- UNVOICED class comprises all unvoiced speech frames and all frames without active speech. A voiced offset frame can be also classified as UNVOICED if its end tends to be unvoiced and the concealment designed for unvoiced frames can be used for the following frame in case it is lost.

- UNVOICED TRANSITION class comprises unvoiced frames with a possible voiced onset at the end. The onset is however still too short or not built well enough to use the concealment designed for voiced frames. The UNVOICED TRANSITION class can follow only a frame classified as UNVOICED or UNVOICED TRANSITION.

- VOICED TRANSITION class comprises voiced frames with relatively weak voiced characteristics. Those are typically voiced frames with rapidly changing characteristics (transitions between vowels) or voiced offsets lasting the whole frame. The VOICED TRANSITION class can follow only a frame classified as VOICED TRANSITION, VOICED or ONSET.

- VOICED class comprises voiced frames with stable characteristics. This class can follow only a frame classified as VOICED TRANSITION, VOICED or ONSET.

- ONSET class comprises all voiced frames with stable characteristics following a frame classified as UNVOICED or UNVOICED TRANSITION. Frames classified as ONSET correspond to voiced onset frames where the onset is already sufficiently well built for the use of the concealment designed for lost voiced frames. The concealment techniques used for a frame erasure following the ONSET class are the same as following the VOICED class. The difference is in the recovery strategy. If an ONSET class frame is lost (i.e. a VOICED good frame arrives after an erasure, but the last good frame before the erasure was UNVOICED), a special technique can be used to artificially reconstruct the lost onset. The artificial onset reconstruction techniques will be described in more detail in the decoder description. On the other hand if a good ONSET frame arrives after an erasure and the last good frame before the erasure was UNVOICED, this special processing is not needed, as the onset has not been lost (has not been in the lost frame).

The classification state diagram is outlined in Figure 5.22-1. The classification information is transmitted using 2 bits. As it can be seen from Figure 5.22-1, UNVOICED TRANSITION class and VOICED TRANSITION class can be grouped together as they can be unambiguously differentiated at the decoder (UNVOICED TRANSITION can follow only UNVOICED or UNVOICED TRANSITION frames, VOICED TRANSITION can follow only ONSET, VOICED or VOICED TRANSITION frames).

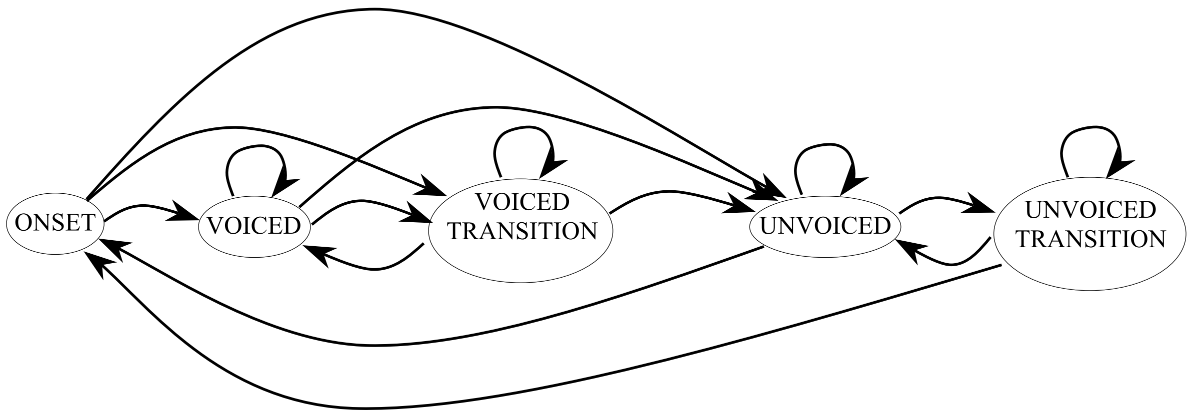

Figure 5.22-1: Block diagram of a frame classification state machine for the erasure concealment.

The following parameters are used for the classification: a normalized correlation $\overline{R}'_{xy}$, a spectral tilt measure e'_t, a signal to noise ratio *snr*, a pitch stability counter *pc*, a relative frame energy of the signal at the end of the current frame E_{rel} and a zero-crossing counter *zc*. As can be seen in the following detailed analysis, the computation of these parameters uses the available look-ahead as much as possible to take into account the behavior of the speech signal also in the following frame.

The average normalized correlation $\overline{R}'_{xy}$ is computed as part of the open-loop pitch search module. It consists of averaging the normalized correlations of the second half-frame and the lookahead. That is

$$\overline{R}'_{xy} = 0.5(C_{norm}(d_1) + C_{norm}(d_2)). \qquad (5.22.1\text{-}1)$$

The spectral tilt parameter e'_t contains the information about the frequency distribution of energy. As described above, the spectral tilt for one spectral analysis is estimated as a ratio between the energy concentrated in low frequencies and the energy concentrated in high frequencies. Here, the tilt measure used is the average in the logarithmic domain of the spectral tilt measures $e_{tilt}(0)$ and $e_{tilt}(1)$ defined in Equation (5.10.2-6). That is,

$$e'_t = 10\log(e_{tilt}(0)e_{tilt}(1)) \qquad (5.22.1\text{-}2)$$

The signal to noise ratio (SNR) measure exploits the fact that for a general waveform matching encoder, the SNR is much higher for voiced sounds. The *snr* parameter estimation must be done at the end of the encoder subframe loop and is computed using the relation

$$snr = \frac{E_{sw}}{E_e} \qquad (5.22.1\text{-}3)$$

where E_{sw} is the energy of the weighted speech signal $\mathbf{s}_w(n)$ of the current frame and E_e is the energy of the error between this weighted speech signal and the weighted synthesis signal of the current frame.

The pitch stability counter *pc* assesses the variation of the pitch period. It is computed as follows:

$$pc = |d_1 - d_0| + |d_2 - d_1| \quad (5.22.1\text{-}4)$$

The values d_0, d_1, and d_2 correspond to the open-loop pitch estimates from the first half of the current frame, the second half of the current frame and the lookahead, respectively.

The last parameter is the zero-crossing parameter *zc* computed on a 20 ms segment of the speech signal. The segment starts in the middle of the current frame and uses two subframes of the lookahead. Here, the zero-crossing counter *zc* counts the number of times the signal sign changes from positive to negative during that interval.

To make the classification more robust, the classification parameters are considered together forming a function of merit *fm*. For that purpose, the classification parameters are first scaled between 0 and 1 so that each parameter's value typical for unvoiced signal translates in 0 and each parameter's value typical for voiced signal translates into 1. A linear function is used between them. The scaled version p^s of a certain parameter p_x is obtained using

$$p^s = k_p p_x + c_p \qquad \text{constrained by} \quad 0 \le p^s \le 1 \quad (5.22.1\text{-}5)$$

The function coefficients k_p and c_p have been found experimentally for each of the parameters so that the signal distortion due to the concealment and recovery techniques used in presence of frame errors is minimal. The values used are summarized in Table 5.22-1.

Table 5.22-1: Signal Classification Parameters and the coefficients of their respective scaling functions

Parameter	Meaning	k_p	c_p
$\overline{R}'_{xy}$	Normalized Correlation	2.857	-1.286
e'_t	Spectral Tilt	0.04167	0
snr	Signal to Noise Ratio	0.1111	-0.3333
pc	Pitch Stability counter	-0.07143	1.857
E_{rel}	Relative Frame Energy	0.05	0.45
zc	Zero Crossing Counter	-0.04	2.4

The merit function has been defined as

$$f = \frac{1}{7}(2\overline{R}^s_{xy} + e^s_t + snr^s + pc^s + E^s_{rel} + zc_s) \quad (5.22.1\text{-}6)$$

where the superscript *s* indicates the scaled version of the parameters.

A first classification decision is made for UNVOICED classes as follows:

If (local_VAD=0) OR (E_{rel}<-8) then class = UNVOICED.

If the above condition is not satisfied, then the classification proceeds using the merit function f_m and following the rules summarized in Table 5.22-2.

Table 5.22-2: Signal Classification Rules at the Encoder

Previous Frame Class	Rule	Current Frame Class
ONSET VOICED VOICED TRANSITION	$f_m \geq 0.66$	VOICED
	$0.66 > f_m \geq 0.49$	VOICED TRANSITION
	$f_m < 0.49$	UNVOICED
UNVOICED TRANSITION UNVOICED	$f_m > 0.63$	ONSET
	$0.63 \geq f_m > 0.585$	UNVOICED TRANSITION
	$f_m \leq 0.585$	UNVOICED

The class information is encoded with two bits as explained above. Despite the fact that the supplementary information, which improves frame erasure concealment, is transmitted only in Generic FR frames, the classification is done for each frame. This is needed to maintain the classification state machine up to date as it uses the information about the previous frame class. The classification is however straightforward for encoding types dedicated to unvoiced or voiced frames. Hence, a VOIVED HR frame is always classified as voiced frame and UNVOICED HR and QR frames are always classified as unvoiced frames.

5.22.2 Other Speech Parameters for Frame Error Processing

In addition to the signal classification information, the other transmitted parameters are energy information and phase information. A precise control of the speech energy is very important in frame error concealment. The importance of the energy control becomes more evident when a normal operation is resumed after an erased block of frames. Since all encoding schemes of active speech make use of a prediction, the actual energy cannot be properly estimated at the decoder. In voiced speech segments, the incorrect energy can persist for several consecutive frames, which can be very annoying especially when this incorrect-valued energy increases.

Although the energy control is very important for voiced speech because of the long-term prediction (pitch prediction), it is also important for unvoiced speech. The reason is the prediction of the innovation gain quantizer. An incorrect value of energy during unvoiced segments can cause an annoying high frequency fluctuation. The phase control can be done in several ways, mainly depending on the available bandwidth. Here, a simple phase control is achieved during lost voiced onsets by searching the approximate information about the glottal pulse position. Hence, apart from the signal classification information discussed in the previous section, the most important information to send is the information about the signal energy and the position of the first glottal pulse in a frame (phase information).

5.22.2.1 Energy Information

The energy information is estimated and sent in the speech signal domain using 6 bits. The energy information is the maximum of the signal energy for frames classified as VOICED or ONSET, or the average energy per sample for other frames. For VOICED or ONSET frames, the maximum signal energy is computed pitch synchronously at the end of the frame as follow:

$$E = \max(\hat{s}^2(i)) \qquad i = L - t_E, \ldots, L-1 \tag{5.22.2.1-1}$$

where L=256 is the frame length and signal $\hat{s}(i)$ is the local synthesis signal. If the pitch delay is greater than 63 samples, t_E equals the rounded closed-loop pitch lag of the last subframe. If the pitch delay is less than 64 samples, then t_E is set to twice the rounded close-loop pitch lag of the last subframe.

For other classes, E is the average energy per sample of the second half of the current frame, i.e. t_E is set to L/2 and the E is computed as:

$$E = \frac{1}{t_E} \sum_{i=L-t_E}^{L-1} \hat{s}^2(i) \qquad (5.22.2.1\text{-}2)$$

The energy information is quantized using a 6 bit uniform quantizer in the range of 0 dB to 96 dB with a step of 1.55 dB. The quantization index is given by the integer part of

$$i = \frac{10\log(E + 0.001)}{1.55} \qquad (5.22.2.1\text{-}3)$$

The index i is then limited to the range $0 \le i \le 63$ and $i \ne 62$. Thus for values of $i < 0$, i is set to 0, and for values $i \ge 62$, i is set to 63. The index pattern $i=62$ (binary '111110') is reserved for the identification of the interoperable encoding types.

5.22.2.2 Phase Control Information

The phase control is particularly important while recovering after a lost segment of voiced speech for similar reasons as described in the previous section. After a block of erased frames, the decoder memories become unsynchronized with the encoder memories. Sending some phase information helps in re-synchronizing the decoder. The rough position of the first glottal pulse in the frame is sent. This information is then used for the recovery after lost voiced onsets as will be described later.

Let T_{sf1} be the rounded closed-loop pitch lag for the first subframe. The position of the first glottal pulse is searched among the T_{sf1} first samples of the frame by looking for the sample with the maximum amplitude. Best results are obtained when the position of the first glottal pulse is measured on the low-pass filtered residual signal. A simple FIR low-pass filter with coefficients 0.25, 0.5 and 0.25 is used.

The position of the first glottal pulse τ is encoded using 6 bits in the following manner. The precision used to encode the position of the first glottal pulse depends on the closed-loop pitch value for the first subframe T_{sf1}. This is possible because this value is known both by the encoder and the decoder, and is not subject to error propagation after one or several frame losses. When T_{sf1} is less than 64, the position of the first glottal pulse relative to the beginning of the frame is encoded directly with a precision of one sample. When $64 \le T_{sf1} < 128$, the position of the first glottal pulse relative to the beginning of the frame is encoded with a precision of two samples by using a simple integer division, i.e., $\tau/2$. When $T_{sf1} \ge 128$, the position of the first glottal pulse relative to the beginning of the frame is encoded with a precision of four samples by further dividing τ by 2. The inverse procedure is done at the decoder. If $T_{sf1} < 64$, the received quantized position is used as is. If $64 \le T_{sf1} < 128$, the received quantized position is multiplied by 2 and incremented by 1. If $T_{sf1} \ge 128$, the received quantized position is multiplied by 4 and incremented by 2 (incrementing by 2 results in uniformly distributed quantization error).

5.23 Encoding of Inactive Speech Frames (CNG-ER and CNG-QR)

Routine Name: `CNG_enc, CNG_synthesis`
Inputs:

- $E(16)$: LP residual energy
- $\overline{N}_f$: Long-term average noise energy
- $\hat{q}_i$: The quantized immitance spectral pairs for current frame

Outputs:

- $\hat{s}(n)$: The synthesized speech signal
- $\overline{q}_i$: The smoothed immitance spectral pairs
- $\overline{E}_s$: The smoothed excitation energy

Initialization:

- Initially, $\overline{E}_s$ is set to $\hat{E}_s$ if not stated otherwise in the following section. $\overline{q}_i$ are initially set to $\hat{q}_i$. The seed value of the excitation random generator is initially set to 21845. Otherwise, buffers and filter memories are set to zero at initialization.

In VMR-WB modes 0, 1, and 2, inactive speech frames are encoded using CNG-ER encoding type. Comfort noise generation (CNG) is used at the decoder to regenerate inactive speech frames. This encoding type is referred to as CNG-ER. In VMR-WB mode 3, due to the interoperability with AMR-WB, encoding of inactive speech frames is performed using the same quantization procedure as in AMR-WB. This requires 35 bits/frame, which does not fit into an ER frame. Note that the last CNG bit is not used in VMR-WB and it is set to 0. Thus a Quarter-Rate frame is used. Encoding of inactive speech frames in VMR-WB mode 3 is referred to as CNG-QR encoding type. In either case, the background noise encoding parameters consist of spectral shape and energy information (LP parameters and excitation energy). At the decoder, CNG is performed by generating a random excitation scaled by a proper gain to drive the LP synthesis filter. The synthesis model uses smoothed LP filter and gain parameters.

5.23.1 LP Parameter Quantization in CNG-ER and CNG-QR

The mean-removed ISF vector at frame *n*, $\mathbf{z}(n)$, is directly quantized without prediction using single-stage split-VQ. In case of CNG-ER, the vector is split into 3 subvectors of dimension 3, 5, and 8, and quantized with 5, 5, and 4 bits, respectively. In the case of CNG-QR, the AMR-WB SID_UPDATE frame quantizer is used whereby the ISF vector is split into 5 subvectors of dimension 2, 3, 3, 4, and 4, and quantized with 6, 6, 6, 5 and 5 bits, respectively. This is shown in Table 5.23-1 and Table 5.23-2.

Table 5.23-1: Quantization of ISF vector for CNG-ER

Unquantized 16-element-long ISF vector		
Split 1 (0-2) 5 bits	Split 2 (3-7) 5 bits	Split 3 (8-15) 4 bits

Table 5.23-2: Quantization of ISF vector for CNG-QR

Unquantized 16-element-long ISF vector				
Split 1 (0-1) 6 bits	Split 2 (2-4) 6 bits	Split 3 (5-7) 6 bits	Split 4 (8-11) 5 bits	Split 4 (12-15) 5 bits

The cdma2000 system does not allow generation of all-zeros and all-ones frames. To comply with this requirement in CNG-ER encoding type, all-zero and all-one patterns for the three subvectors are prevented. If the quantizer chooses an all-one index for the first two subvectors (index 31), then in the quantization of the third subvector, the last index is not permitted. Similarly, if the quantizer chooses an all-zero index for the first two subvectors (index 0), then in the quantization of the third subvector, the first two indices (indices 0 and 1) are not permitted. Index 0 is not permitted to prevent all zero frames, and index 1 is not permitted since the pattern '00000000000001' is reserved to signal NO_DATA frames. This is useful for interoperability between AMR-WB and VMR-WB when AMR-WB is operating in DTX mode (see Section 7).

After quantization in the frequency domain, the ISF parameters are converted into the cosine domain to obtain the ISP vector $\hat{\mathbf{q}}$.

5.23.2 Energy Quantization in CNG-ER and CNG-QR

The energy per sample is computed from the LP residual energy. The LP residual energy is a byproduct of the LP analysis procedure and it is given by the value *E*(16) in the Levinson-Durbin recursion of Equation (5.6.2-2). The energy per sample is computed as

$$E_s = 0.0059322\ E(16) \tag{5.23.2-1}$$

where the scaling of E(16) takes into account the LP analysis window size and shape. In case of the first inactive speech frame after an active speech frame and very low background noise ($\overline{N}_f$ <16), E_s is multiplied by 0.1 to prevent smearing of the respiration noise. The energy per sample is then converted to the $\log_2$ domain for quantization purposes as

$$E_{s,2} = \log_{10}(E_s)/\log_{10}(2) \tag{5.23.2-2}$$

For narrowband signals, $E_{s,2}$ is increased by 0.5 to compensate for limited bandwidth of the input signal. The computed energy in the $\log_2$ domain is then offset by a certain negative value E_{offset} in the range [$E_{offset_{\min}}$,0] where $E_{offset_{\min}} = -3$ for narrowband inputs and $E_{offset_{\min}} = -2.2032$ for wideband inputs (corresponding approximately to -9 dB and –6.6 dB, respectively).

For the first inactive frame after an active speech frame, the offset value is computed as

$$E_{offset} = 0.1102\overline{N}_f - 3.8556 \text{ constrained by } E_{offset_{\min}} \le E_{offset} \le 0 \tag{5.23.2-3}$$

where $\overline{N}_f$ is the long-term average noise energy given in Equation (5.4.3-2). In subsequent inactive speech frames, E_{offset} converges to $E_{offset_{\min}}$ using the relation

$$E_{offset} = 0.9E_{offset} + 0.1E_{offset_{\min}} \tag{5.23.2-4}$$

The offset energy in the $\log_2$ domain is given by

$$E_{s,2} = E_{s,2} + E_{offset} \tag{5.23.2-5}$$

and it is quantized using 6 bits. A uniform quantizer is used with level 0 at -2 and level 63 at 22, and with quantization step 24/63. The quantization index is found using the relation

$$index = \left\lfloor \left((E_{s,2} + 2)\frac{63}{24} \right) \right\rfloor \tag{5.23.2-6}$$

and the quantized value is found by

$$\hat{E}_{s,2} = \frac{24}{63} index - 2 \tag{5.23.2-7}$$

The quantized energy per sample in the linear domain is then found as

$$\hat{E}_s = 2^{\hat{E}_{s,2}} \quad (5.23.2\text{-}8)$$

5.23.3 Local CNG Synthesis

Local CNG synthesis is performed at the encoder in order to update the filters and adaptive codebook memories. CNG is performed by generating 20 ms random values, scaling by a gain computed from the smoothed quantized energy, and filtering the scaled excitation through a smoothed LP synthesis filter. Random short integer values are generated using the relation

$$seed = short(seed \times 31821 + 13849) \quad (5.23.3\text{-}1)$$

with the seed value initially set to 21845. The normalized random excitation sequence (with energy per sample equal to 1) is computed from the random integer sequence $r_n(n)$ for the 256-sample frame using the relation

$$c(n) = r_n(n)\sqrt{256 / \sum_{i=0}^{255} r_n(n)} \quad (5.23.3\text{-}2)$$

For the first inactive speech frame after an active speech frame, the smoothed energy used for synthesis is updated as

$$\overline{E}_s = 0.5\overline{E}_s + 0.5\hat{E}_s \quad (5.23.3\text{-}3)$$

with the exception of incoming NO_DATA frames from AMR-WB encoder in the AMR-WB interoperable mode (Section 7). The first NO_DATA frame is usually preceded by an inactive speech frame encoded at Full-Rate (i.e., AMR-WB VAD bit set to 0). In this case, the mean energy of the adaptive codebook memory is used for the update of $\overline{E}_s$ with the weighting of 0.2 and 0.8 for the mean energy of the adaptive codebook and $\overline{E}_s$ respectively. If the AMR-WB NO_DATA frame is not preceded by an inactive Full-Rate frame, no $\overline{E}_s$ update is performed. For consequent frames, the smoothed energy is updated as

$$\overline{E}_s = 0.7\overline{E}_s + 0.3\hat{E}_s \quad (5.23.3\text{-}4)$$

Initially, $\overline{E}_s$ is set equal to $\hat{E}_s$. In the AMR-WB interoperable mode (VMR-WB mode 3), $\overline{E}_s$ is initially set to the mean energy of the adaptive codebook memory in case of a NO_DATA frame. The energy scaled excitation signal is given by

$$u(n) = \sqrt{\overline{E}_s}\, c(n)\text{, n=0,…,255} \quad (5.23.3\text{-}5)$$

The smoothed ISP vectors are updated as

$$\overline{\mathbf{q}} = 0.9\overline{\mathbf{q}} + 0.1\hat{\mathbf{q}} \quad (5.23.3\text{-}6)$$

where initially $\overline{\mathbf{q}}$ is set equal to $\hat{\mathbf{q}}$. The processing is once again different for the first inactive ER or QR speech frame after an active speech. If such a frame is a NO_DATA frame preceded by an inactive Full-Rate frame, the update procedure described above is used with weighting of 0.8 and 0.2. For other first ER or QR inactive frames $\overline{\mathbf{q}}$ is not updated. The smoothed ISP parameter vector is converted to the LP coefficient domain to obtain the LP synthesis filter. The comfort noise is generated by filtering the scaled excitation $u(n)$ through the smoothed synthesis filter.

5.23.4 Memory Update in CNG-ER and CNG-QR

After synthesizing inactive speech frames using CNG as described in the previous section, the memory of the synthesis filter and adaptive codebook are updated. In the AMR-WB interoperable mode, the adaptive codebook is reset. The buffers used in signal modification are also updated. The previous frame ISPs $\hat{\mathbf{q}}_4^{(n-1)}$ are updated with $\overline{\mathbf{q}}$. Other memory parameters are reset to their initial states (e.g., ISP and gain quantizers, weighting filter, etc.).

Note that encoding of CNG-ER and CNG-QR types is performed directly after the rate selection described in Section 5.10. If either CNG-ER or CNG-QR encoding type is chosen then the processing described in this section is performed and no further processing related to the other encoding types is further performed.

6 FUNCTIONAL DESCRIPTION OF THE DECODER

The function of the decoder consists of decoding the transmitted parameters (LP parameters, adaptive codebook vector, adaptive codebook gain, fixed-codebook vector, fixed-codebook gain) and performing synthesis to obtain the reconstructed speech. The signal flow at the decoder is shown in Figure 4.1-3.

The decoding process starts with decoding the LP filter parameters. The received indices of ISP quantization are used to reconstruct the quantized ISP vector. The interpolation described in Section 5.6.5 is performed to obtain 4 interpolated ISP vectors (corresponding to 4 subframes). Then the excitation signal is reconstructed and post-processed before performing LP synthesis filtering to obtain the reconstructed speech. The reconstructed speech is then de-emphasized (inverse of pre-emphasis applied at the encoder). Finally, a post-processing is applied for enhancing the periodicity in the low frequency region of the signal, and then the signal is up-sampled to 16 kHz. Finally high-band signal is generated to the frequency band from 6 to 7 kHz. Both parts of the signal are added to obtain the full-band reconstructed speech.

More details about the excitation reconstruction and excitation post-processing, synthesis and post-processing of synthesis signal, and high frequency generation will be provided in the following sections.

6.1 Reconstruction of the Excitation

The decoding process is performed in the following order: For each subframe, the interpolated ISP vector is converted to LP filter coefficient domain a_k, which is used for synthesizing the reconstructed speech in the subframe.

The following steps are repeated for each subframe:

1) **Decoding of the adaptive codebook vector**: In case of FR and Generic HR encoding types, the received pitch index (adaptive codebook index) is used to find the integer and fractional parts of the pitch lag. The initial adaptive codebook excitation vector $v'(n)$ is found by interpolating the past excitation $u(n)$ (at the pitch delay) using the FIR filter described in Section 5.16. In case of Generic HR no low-pass filtering is used and the adaptive codebook excitation is $v(n) = v'(n)$. In case of FR, the received adaptive filter index is used to decide whether the filtered adaptive codebook is $v(n) = v'(n)$ or $v(n) = 0.18v'(n) + 0.64v'(n-1) + 0.18v'(n-2)$. In case of Voiced HR, the delay contour is first computed for the whole frame as described in Equation (5.11.2-1). The initial adaptive codebook excitation $v'(n)$ in certain subframes is computed by interpolating the past excitation in the adaptive codebook buffer at the delays given by the delay contour as described in Section 5.16.4. The received adaptive filter index is then used to find out whether the filtered adaptive codebook is $v(n) = v'(n)$ or $v(n) = 0.18v'(n) + 0.64v'(n-1) + 0.18v'(n-2)$. In case of Unvoiced encoding types and CNG, there is no adaptive codebook contribution.

2) **Decoding of the innovative vector**: In case of FR, Voiced HR, and Generic HR, the received algebraic codebook index is used to extract the positions and amplitudes (signs) of the excitation pulses and to find the algebraic codevector $c(n)$. If the integer part of the pitch lag is less than the subframe size 64, the pitch sharpening procedure is applied, which translates into modifying $c(n)$ by filtering it through the adaptive pre-filter $F(z)= (1 - \beta_1 z^{-1})/ (1-0.85z^{-T})$ which further consists of two parts: a periodicity enhancement part $1/(1-0.85z^{-T})$, where T is the integer part of the pitch lag representing the fine spectral structure of the speech signal, and a tilt part $(1 - \beta_1 z^{-1})$, where β_1 is related to the voicing of the previous subframe and is bounded by [0.0,0.5]. The periodicity enhancement part of the filter colors the spectrum by damping inter-harmonic frequencies, which are annoying to the human ear

in case of voiced signals. In case of FR and Generic HR encoding types, T is the integer part of the pitch lag. In case of Voiced HR, T is computed on a sample basis following the pitch contour.

In case of Unvoiced HR, the signs and indices of the two random vectors are decoded and the excitation is reconstructed as in Equation (5.18.1-1). In case of Unvoiced QR, the excitation is reconstructed as in Equation (5.18.1-1) but with randomly chosen indices and signs.

3) **Decoding of the adaptive and innovative codebook gains**: In FR, Voiced HR, and Generic HR encoding types, the received index provides the adaptive codebook gain $\hat{g}_p$ and the fixed-codebook gain correction factor $\hat{\gamma}$. Note that in case of Voiced HR and Generic HR, only half of the gain codebook is used. The estimated fixed-codebook gain g'_c is found as described in Section 5.20. First, the predicted energy for every subframe n is found by

$$\widetilde{E}(n) = \sum_{i=1}^{4} b_i \hat{R}(n-i) \tag{6.1-1}$$

and then the average innovation energy is found by

$$E_i = 10\log\left(\frac{1}{N}\sum_{i=0}^{N-1} c^2(i)\right) \tag{6.1-2}$$

the predicted gain G'_c in dB is found by

$$G'_c = \widetilde{E}(n) + \overline{E} - E_i \tag{6.1-3}$$

The prediction gain in the linear domain is given by

$$g'_c = 10^{0.05G_c} = 10^{0.05(\widetilde{E}(n)+\overline{E}-E_i)} \tag{6.1-4}$$

The quantized fixed-codebook gain is given by

$$\hat{g}_c = \hat{\gamma} g'_c \tag{6.1-5}$$

In case of Unvoiced HR and Unvoiced QR, only the fixed-codebook gain is transmitted. The received index k gives the gain prediction error in dB, $\hat{\Gamma}$, using the relation

$$\hat{\Gamma} = k \times \delta + \Gamma_{\min} \tag{6.1-6}$$

with the quantization step defined in Equation (5.20.1-2) and with the values $\Gamma_{\min}$ and δ for Unvoiced HR and Unvoiced QR given in Section 5.20.1. The predicted gain is then computed as in (6.1-3) and the quantized gain in dB is found by

$$\hat{G}_c = \hat{\Gamma} + G'_c \quad (6.1\text{-}7)$$

and the quantized gain is given by

$$\hat{g}_c = 10^{0.05\hat{G}_c} \quad (6.1\text{-}8)$$

4) **Computing the reconstructed excitation**: The following steps are for *n* = 0, ..., 63. The total excitation is constructed by:

$$u'(n) = \hat{g}_p v(n) + \hat{g}_c c(n) \quad (6.1\text{-}9)$$

where *c*(*n*) is the codevector from the fixed-codebook after filtering it through the adaptive pre-filter *F*(*z*). The excitation signal u'(n) is used to update the content of the adaptive codebook. The excitation signal u'(n) is then post-processed as described in the next section to obtain the post-processed excitation signal u(n) used at the input of the synthesis filter $1/\hat{A}(z)$.

6.2 Excitation Post-processing

Routine Name: `enhancer, est_tilt, isf_stab`
Inputs:

- *v*(*n*): The adaptive codevector
- *c*(*n*): The algebraic codevector
- $\hat{g}_p$: The quantized adaptive codebook gain
- $\hat{g}_c$: The quantized algebraic codebook gain
- f_i : The immitance spectral frequencies for current frame
- $f_i^{(p)}$: The immitance spectral frequencies for previous frame

Outputs:

- *u*(*n*): The enhanced excitation signal
- θ: The ISF stability factor
- r_v : The tilt of the excitation

Initialization:

- Memories and buffers are initialized to zero.

Before speech synthesis, a post-processing of excitation elements is performed.

6.2.1 Anti-Sparseness Processing in Generic HR

An adaptive anti-sparseness post-processing procedure is applied to the fixed-codebook vector *c*(*n*) in the Generic HR encoding type in order to reduce perceptual artifacts arising from the sparseness of the algebraic fixed-codebook vectors with only a few non-zero samples per subframe. The anti-sparseness processing consists of circular convolution of the fixed-codebook vector with an impulse response. Three pre-stored impulse responses are used and a number *impNr* = 0,1, and 2 is set to select one of them. A value of 2 corresponds to no modification; a value of 1 corresponds to medium modification, while a value of 0 corresponds to strong modification. The selection of the impulse response is performed adaptively from the adaptive and fixed-codebook gains. The following procedure is employed:

$$\begin{aligned}
&\text{if } \hat{g}_p < 0.6 \text{ then}\\
&\quad impNr = 0;\\
&\text{else if } \hat{g}_p < 0.9 \text{ then}\\
&\quad impNr = 1;\\
&\text{else}\\
&\quad impNr = 2;
\end{aligned}$$

In other words, the onset is detected by comparing the fixed-codebook gain to the previous fixed-codebook gain. If the current value is more than three times the previous value an onset is detected.

If the onset is not detected and *impNr* = 0, the median filtered value of the current and the previous 4 adaptive codebook gains are computed. If this value is less than 0.6, *impNr* = 0.

Also, if the onset is not detected, the *impNr* value is restricted to increase by one step from the previous subframe.

If an onset is detected, the *impNr* value is increased by one, if it is less than 2.

6.2.2 Gain Smoothing for Noise Enhancement

A nonlinear gain smoothing technique is applied to the fixed-codebook gain $\hat{g}_c$ in order to enhance excitation in noise. Based on the stability and voicing of the speech segment, the gain of the fixed-codebook vector is smoothed in order to reduce fluctuation in the energy of the excitation in case of stationary signals. This improves the performance in case of stationary background noise. The voicing factor is given by

$$\lambda = 0.5(1 - r_v) \tag{6.2.2-1}$$

with

$$r_v = (E_v - E_c)/(E_v + E_c), \tag{6.2.2-2}$$

where E_v and E_c are the energies of the scaled pitch codevector and scaled innovation codevector, respectively (r_v gives a measure of signal periodicity) . Note that since the value of r_v is between –1 and 1, the value of λ is between 0 and 1. Note that the factor λ is related to the amount of unvoicing with a value of 0 for purely voiced segments and a value of 1 for purely unvoiced segments.

A stability factor θ is computed based on a distance measure between the adjacent LP filters. Here, the factor θ is related to the ISF distance measure. The ISF distance is given by

$$ISF_{dist} = \sum_{i=0}^{14} (f_i - f_i^{(p)})^2 \tag{6.2.2-3}$$

where f_i are the ISFs in the present frame, as defined in Equation (5.13-1), and $f_i^{(p)}$ are the ISFs in the past frame. The stability factor θ is given by

$$\theta = 1.25 - ISF_{dist} / 400000 \quad \text{Constrained by } 0 \le \theta \le 1 \tag{6.2.2-4}$$

The ISF distance measure is smaller in case of stable signals. As the value of θ is inversely related to the ISF distance measure, then larger values of θ correspond to more stable signals. The gain-smoothing factor S_m is given by

$$S_m = \lambda\theta \tag{6.2.2-5}$$

The value of S_m approaches 1 for unvoiced and stable signals, which is the case of stationary background noise signals. For purely voiced signals or for unstable signals, the value of S_m approaches 0. An initial modified gain g_0 is computed by comparing the fixed-codebook gain $\hat{g}_c$ to a threshold given by the initial modified gain from the previous subframe, g_{-1}. If $\hat{g}_c$ is larger or equal to g_{-1}, then g_0 is computed by decrementing $\hat{g}_c$ by 1.5 dB bounded by $g_0 \ge g_{-1}$. If $\hat{g}_c$ is smaller than g_{-1}, then g_0 is computed by incrementing $\hat{g}_c$ by 1.5 dB constrained by $g_0 \le g_{-1}$.

Finally, the gain is updated with the value of the smoothed gain as follows

$$\hat{g}_c = S_m g_0 + (1 - S_m)\hat{g}_c \tag{6.2.2-6}$$

6.2.3 Pitch Enhancer for Generic and Voiced Encoding Types

A pitch enhancer scheme modifies the total excitation $u'(n)$ by filtering the fixed-codebook excitation through an innovation filter whose frequency response emphasizes the higher frequencies and reduces the energy of the low frequency portion of the innovative codevector, and whose coefficients are related to the periodicity in the signal. A filter of the form

$$F_{inno}(z) = -c_{pe} z + 1 - c_{pe} z^{-1} \tag{6.2.3-1}$$

is used where $c_{pe} = 0.125(1 - r_v)$, with r_v being a periodicity factor given by $r_v = (E_v - E_c)/(E_v + E_c)$ as described above. The filtered fixed-codebook codevector is given by

$$c'(n) = c(n) - c_{pe}(c(n+1) + c(n-1)) \tag{6.2.3-2}$$

and the updated post-processed excitation is given by

$$u(n) = \hat{g}_p v(n) + \hat{g}_c c'(n) \tag{6.2.3-3}$$

The above procedure can be done in one step by updating the excitation as follows

$$u(n) = u'(n) - \hat{g}_c c_{pe}(c(n+1) + c(n-1)) \tag{6.2.3-4}$$

6.3 Synthesis, Post-processing and Up-sampling

Routine Name: `deemph, bass_postfilter`
Inputs:

- $u'(n)$: The excitation signal
- $\hat{g}_p$: The quantized adaptive codebook gain
- $\hat{a}_i$: The quantized LP filter coefficients
- T: Transmitted pitch lag values

Outputs:

- $\hat{s}(n)$: The lower band speech synthesis at 12.8 kHz sampling rate
- The 16 or 8 kHz sampled synthesized speech.

Initialization:

- All memories and buffers are set to zero at initialization except the mean prediction error energy $\overline{E}_{pp}$, which is set to 40 at initialization.

The synthesis is performed by filtering the post-processed excitation signal $u(n)$ through the LP synthesis filter $1/\hat{A}(z)$. The synthesized signal is then de-emphasized by filtering through the filter $1/(1-0.68z^{-1})$ (inverse of the pre-emphasis filter applied at the input). The synthesis and de-emphasis modules reproduce the reconstructed signal in the down-sampled domain of 12.8 kHz. The signal needs to be up-sampled to 16 kHz and then added to the high frequency signal generated from 6.4 to 8 kHz as will described below. Before up-sampling to 16 kHz, low frequency pitch enhancement is applied to the synthesized signal and this is combined with the up-sampling procedure as will be described in the next section.

6.3.1 Low-Frequency Pitch Enhancement Post-processing

In the low-frequency pitch enhancement, two-band decomposition is used and adaptive filtering is applied only to the lower band. This results in a total post-processing that is mostly targeted at frequencies near the first harmonics of the synthesized speech signal.

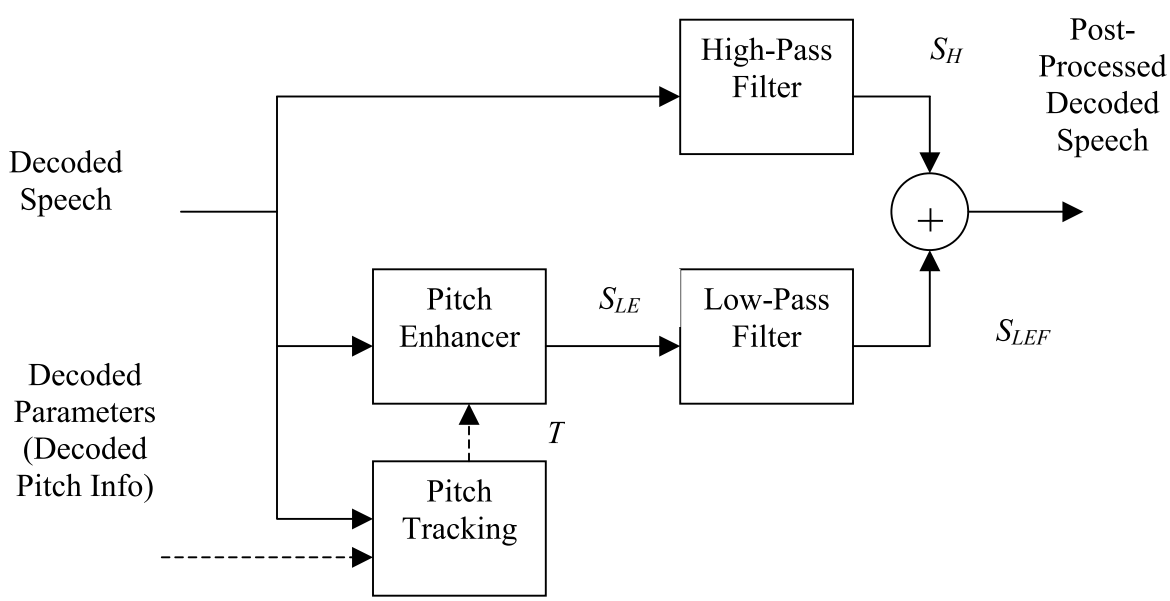

Figure 6.3-1: Block diagram of the low frequency pitch enhancer.

Figure 6.3-1 shows the block diagram of the two-band pitch enhancer. In the higher branch the decoded speech signal is filtered by a high-pass filter to produce the higher band signal (s_H). In the lower branch, the decoded speech signal is first processed through an adaptive pitch enhancer, and then filtered through a low-pass filter to obtain the lower band, post-processed signal (s_{LEF}). The post-processed decoded speech signal is obtained by adding the lower band post-processed signal and the higher band signal. The object of the pitch enhancer is to reduce the inter-harmonic noise in the decoded speech signal, which is achieved here by a time-varying linear filter described by the following equation:

$$\hat{s}_f(n) = (1-\alpha)\hat{s}(n) + \alpha s_p(n) \tag{6.3.1-1}$$

where α is a coefficient that controls the inter-harmonic attenuation, T is the pitch period of the input signal $\hat{s}(n)$ and $\hat{s}_f(n)$ is the output signal of the pitch enhancer. $s_p(n)$ is the two-sided long-term prediction signal that is computed in each subframe as

$$s_p(n) = 0.5\hat{s}(n-T) + 0.5\hat{s}(n+T) \tag{6.3.1-2}$$

Parameters T and α vary with time and are given by the pitch tracking module. With a value of α = 1, the gain of the filter described by Equation (6.3.1-1) is exactly 0 at frequencies $1/(2T)$,$3/(2T)$, $5/(2T)$, etc.; i.e. at the mid-point between the harmonic frequencies $1/T$, $3/T$, $5/T$, etc. When α approaches 0, the attenuation between the harmonics produced by the filter of Equation (6.3.1-1) decreases.

In case of FR and Generic HR encoding types, the received closed-loop pitch lag in each subframe is directly used (the fractional pitch lag rounded to the nearest integer). In case of Voiced HR, the pitch lag in the middle of the subframe is used (based on the delay contour).

The factor α is computed as follows. The correlation between the signal and the predicted signal is given by

$$C_p = \sum_{n=0}^{N-1} \hat{s}(n) s_p(n) \tag{6.3.1-3}$$

and the energy of the predicted signal is given by

$$E_p = \sum_{n=0}^{N-1} s_p(n) s_p(n) \tag{6.3.1-4}$$

The factor α is given by

$$\alpha = \frac{C_p}{0.5(E_p + 10^{0.1\bar{E}_{pp}})} \quad \text{constrained by } 0 \le \alpha \le 1 \tag{6.3.1-5}$$

where $\bar{E}_{pp}$ is the mean prediction error energy in dB in the present subframe. The mean prediction error energy $\bar{E}_{pp}$ is updated for the next subframe as follows. The long-term prediction error is first computed by

$$e_p(n) = \hat{s}(n) - \frac{C_p}{E_p} s_p(n) \tag{6.3.1-6}$$

and then pre-emphasized using the relation

$$e_{pp}(n) = e_p(n) - 0.95e_p(n-1) \qquad (6.3.1\text{-}7)$$

The energy of the pre-emphasized error signal is then computed in dB as

$$E_{pp} = 10\log\left[\sum_{n=0}^{N-1} e_{pp}(n)e_{pp}(n)\right] \qquad (6.3.1\text{-}8)$$

The mean error energy is then updated in every subframe by

$$\overline{E}_{pp} = 0.99\overline{E}_{pp} + 0.01E_{pp} \qquad (6.3.1\text{-}9)$$

with initial value $\overline{E}_{pp} = 40$

Note that in order to perform forward pitch prediction in Equation (6.3.1-2) (to compute $\hat{s}(n+T)$), the synthesis should be extended after the end of the frame by 231 samples (the maximum pitch lag). This is done by first extending the excitation signal using the relation

$$u(n+L) = \hat{g}_p u(n+L-T), \quad n = 0,...,230 \qquad (6.3.1\text{-}10)$$

where *L*=256 is the frame size, and then computing the extended synthesis signal by applying synthesis filtering and de-emphasis.

In the VMR-WB C simulation, since the signal need to be re-sampled to 16 kHz using an interpolation filter, the high-pass and low-pass filters in Figure 6.3-1 are combined with the interpolation filter to directly compute the pitch enhanced re-sampled signal. This is shown in Figure 6.3-2. The low frequency pitch enhancement is applied to the first 500 Hz of the frequency band. In case of up-sampling to 16 kHz, the band-pass filter in Figure 6.3-2 has a bandwidth from 500 Hz to 6.4 kHz in the 32 kHz up-sampled bandwidth. The Low pass filter has a bandwidth of 500 Hz.

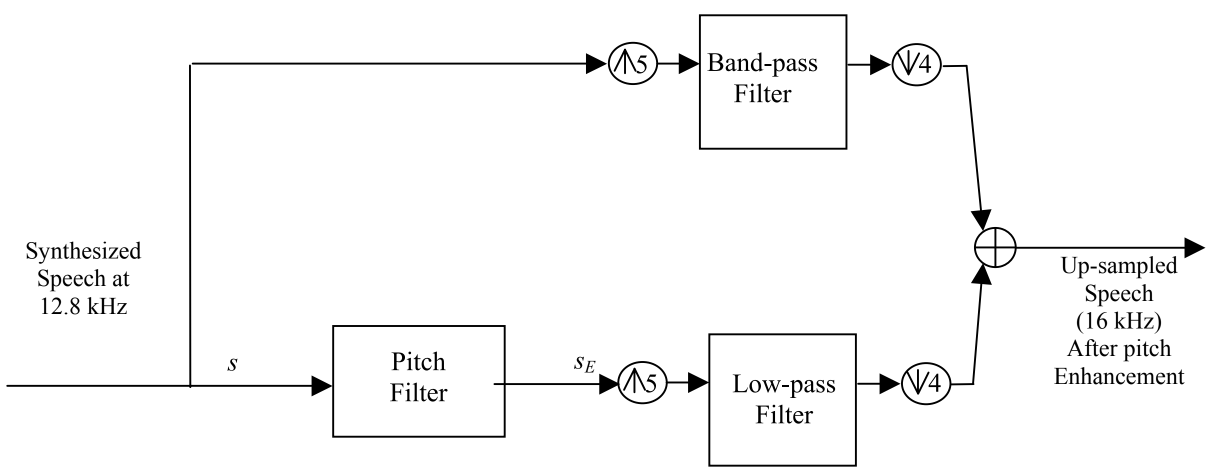

Figure 6.3-2: Block diagram of low frequency pitch enhancement combined with up-sampling.

6.3.2 High-Pass Filtering

After low frequency pitch enhancement and re-sampling, the synthesis signal is high pass filtered as a precaution against undesired low frequency components. At 16 kHz output, a 50-Hz high pass filter is used, in the form

$$H_{hp16k}(z) = \frac{0.986211925 - 1.972423850\mathrm{z}^{-1} + 0.986211925\mathrm{z}^{-2}}{1 + 1.972233729\mathrm{z}^{-1} - 0.972613969\mathrm{z}^{-2}} \quad (6.3.2\text{-}1)$$

At 8 kHz output, a 100-Hz high pass filter is used, in the form

$$H_{hp8k}(z) = \frac{0.945976856 - 1.891953712\mathrm{z}^{-1} + 0.945976856\mathrm{z}^{-2}}{1 + 1.889033079\mathrm{z}^{-1} - 0.894874345\mathrm{z}^{-2}} \quad (6.3.2\text{-}2)$$

6.4 Reconstruction of High-Frequency Band

Routine Name: `hf_synth`

Inputs:

- $\hat{s}(n)$: The lower band speech synthesis at 12.8 kHz sampling rate
- $u(n)$: The enhanced excitation signal
- $\hat{a}_i$: The quantized LP filter coefficients

Outputs:

- $\hat{s}_{16k}(n)$: The 16 kHz sampled synthesized speech with reconstructed high-frequency band

Initialization:

- All buffers and filter memories are set to zero at initialization. The seed of high-frequency random regeneration is initialized to 21845.

For the higher frequency band (6.4-7.0 kHz), excitation is generated to model the highest frequencies. The high frequency content is generated by filling the upper part of the spectrum with a white noise properly scaled in the excitation domain, then converted to the speech domain by spectrally shaping it with a filter derived from the same LP synthesis filter used for synthesizing the down-sampled signal.

6.4.1 Generation of High-Band Excitation

The high-band excitation is obtained by first generating white noise $u_{HB1}(n)$. The power of the high-band excitation is set equal to the power of the lower band excitation $u_2(n)$, which means that

$$u_{HB2}(n) = u_{HB1}(n)\sqrt{\frac{\sum_{k=0}^{63} u_2^{\,2}(k)}{\sum_{k=0}^{63} u_{HB1}^{\,2}(k)}} \quad (6.4.1\text{-}1)$$

Finally the high-band excitation at 16 kHz sampling is found by

$$u_{HB}(n) = \hat{g}_{HB} u_{HB2}(n) \tag{6.4.1-2}$$

where $\hat{g}_{HB}$ is a scaling gain factor estimated using voicing information bounded by [0.1,1.0]. First, tilt of synthesis e_{tilt} is found

$$e_{\text{tilt}} = \frac{\sum_{n=1}^{63} \hat{s}_{hp}(n)\hat{s}_{hp}(n-1)}{\sum_{n=0}^{63} \hat{s}_{hp}{}^2(n)} \tag{6.4.1-3}$$

where $\hat{s}_{hp}(n)$ is high-pass filtered lower band speech synthesis $\hat{s}(n)$ with cut-off frequency of 400 Hz. The scaling gain $\hat{g}_{HB}$ is then found by

$$g_{HB} = w_{SP} g_{SP} + (1 - w_{SP}) g_{BG} \tag{6.4.1-4}$$

where g_{SP} = 1 – e_{tilt} is the gain for speech signal, g_{BG} = 0 is the gain for background noise signal, and w_{SP} is a weighting function set to 1, when VAD is ON, and 0 when VAD is OFF. In other words, g_{HB} = g_{SP} in case of speech signal and g_{HB} =0 in case of background noise signal (no high frequency generation). The VAD information is derived from the encoding type (VAD is OFF in case of CNG-ER or CNG-QR and VAD is ON otherwise). The encoding rate is signaled to the decoder by the receiving site multiplex sublayer. g_{HB} is bounded between [0.1, 1.0]. In case of voiced segments where less energy is present at high frequencies, e_{tilt} approaches 1 resulting in a lower gain g_{HB}. This reduces the energy of the generated noise in case of voiced segments.

6.4.2 LP Filter for the High-Frequency Band

The high-band LP synthesis filter $A_{HB}(z)$ is used for spectral shaping of the high frequency noise signal. The filter $A_{HB}(z)$ is given by the weighted low-band LP synthesis filter as

$$A_{HB}(z) = \hat{A}(z / 0.8) \tag{6.4.2-1}$$

where $\hat{A}(z)$ is the interpolated LP synthesis filter. $\hat{A}(z)$ has been computed analyzing signal with the sampling rate of 12.8 kHz but it is now used for a 16 kHz signal. Effectively, this means that the frequency response $FR_{16}(f)$ of $A_{HB}(z)$ is obtained by

$$FR_{16}(f) = FR_{12.8}\left(\frac{12.8}{16} f\right) \tag{6.4.2-2}$$

where $FR_{12.8}(f)$ is the frequency response of $A(z)$. This means that the band 5.1-5.6 kHz in 12.8 kHz domain will be mapped to 6.4-7.0 kHz in 16 kHz domain.

6.4.3 High-Band Synthesis

$u_{HB}(n)$ is filtered through $A_{HB}(z)$. The output of this high-band synthesis $s_{HB}(n)$ is filtered through a band-pass FIR filter $H_{HB}(z)$ which has the pass-band from 6 to 7 kHz. Finally, s_{HB} is added to the synthesized speech $\hat{s}_{16k}(n)$ to produce the synthesized output speech signal $\hat{s}_{\text{output}}(n)$.

6.5 Frame Error Concealment

Routine Name: `isf_dec_bfi, syn_bfi`

Inputs:

- $u'(n)$: The excitation signal from the previous frame
- Classification decision of previous frame
- θ: The ISF stability factor
- $b(i)$: The adaptive codebook gains for each subframe of last good frame
- $g(i)$: The algebraic codebook gains for each subframe of last good frame
- T_c: Last reliable pitch lag
- r_v : The tilt of the excitation
- $\overline{E}_s$: The smoothed CNG excitation energy

Outputs:

- $u'(n)$: The excitation signal with controlled energy
- $u(n)$: The enhanced excitation signal with controlled energy
- E: The synthesized speech energy

Initialization:

- All buffers and filter memories are set to zero at initialization. The seed used to generate the random part of the excitation is initialized to 21845.

In case of frame erasures, the concealment strategy can be summarized as a convergence of the signal energy and the spectral envelope to the estimated parameters of the background noise. The periodicity of the signal is converged to zero. The speed of the convergence is dependent on the parameters of the last correctly received frame class and the number of consecutive erased frames and is controlled by an attenuation factor α. The factor α is further dependent on the stability of the LP filter for UNVOICED frames. In general, the convergence is slow if the last good received frame is in a stable segment and is rapid if the frame is in a transition segment. The values of α are summarized in Table 6.5-1.

Table 6.5-1: Values of the FER concealment attenuation factor α

Last Good Received Frame	Number of successive erased frames	α
ARTIFICIAL ONSET		0.6
ONSET, VOICED	≤ 3	1.0
	> 3	0.4
VOICED TRANSITION		0.4
UNVOICED TRANSITION		0.8
UNVOICED	= 1	0.6 θ + 0.4
	> 1	0.4

A stability factor θ is computed based on a distance measure between the adjacent LP filters. Here, the factor θ is related to the ISF distance measure and it is bounded by $0 \le \theta \le 1$, with larger values of θ corresponding to more stable signals. This results in decreasing energy and spectral envelope fluctuations when an isolated frame erasure occurs inside a stable unvoiced segment. The signal class remains unchanged during the processing of erased frames, i.e. the class remains the same as in the last correctly received frame.

6.5.1 Construction of the Periodic Part of the Excitation

When a frame is found to be in error by the receiving site multiplex sublayer, an Erasure packet type is provided to the VMR-WB decoder. For a concealment of erased frames following a correctly received UNVOICED frame, no periodic part of the excitation is generated. For a concealment of erased frames following a correctly received frame other than UNVOICED, the periodic part of the excitation is constructed by repeating the last pitch period of the previous frame. If this is the case of the 1st erased frame after a good frame, this pitch pulse is first low-pass filtered. The filter used is a simple 3-tap linear phase FIR filter with the coefficients equal to 0.18, 0.64 and 0.18. The pitch period T_c used to select the last pitch pulse and hence used during the concealment is defined so that pitch multiples or submultiples can be avoided, or reduced. The following logic is used in determining the pitch period T_c

if ((T_3 < 1.8 T_s) AND (T_3 > 0.6 T_s)) OR ($T_{cnt} \geq 30$), then $T_c = T_3$, else $T_c = T_s$.

Here, T_3 is the rounded pitch period of the 4th subframe of the last good received frame and T_s is the rounded pitch period of the 4th subframe of the last good stable voiced frame with coherent pitch estimates. A stable voiced frame is defined here as a VOICED frame preceded by a frame of voiced type (VOICED TRANSITION, VOICED, ONSET). The coherence of pitch is verified in this implementation by examining whether the closed-loop pitch estimates are reasonably close; i.e. whether the ratios between the last subframe pitch, the 2nd subframe pitch and the last subframe pitch of the previous frame are within the interval (0.7, 1.4).

This determination of the pitch period T_c implies that if the pitch at the end of the last good frame and the pitch of the last stable frame are close, the pitch of the last good frame is used. Otherwise, this pitch is considered unreliable and the pitch of the last stable frame is used instead to avoid the impact of erroneous pitch estimates at voiced onsets. This logic is valid only if the last stable segment is not too far in the past. Hence a counter T_{cnt} is defined that limits the effect of the last stable segment. If T_{cnt} is greater or equal to 30; i.e. if there are at least 30 frames since the last T_s update, the last good frame pitch is used systematically. T_{cnt} is reset to 0 every time a stable segment is detected and T_s is updated. The period T_c is then maintained constant during the concealment for the entire erased block.

As the last pulse of the excitation of the previous frame is used for the construction of the periodic part, its gain is approximately correct at the beginning of the concealed frame and can be set to 1. The gain is then attenuated linearly throughout the frame on a sample-by-sample basis to achieve the value of α at the end of the frame.

The values of α correspond to the Table 6.5-1 with the exception that they are modified for erasures following VOICED and ONSET frames to take into account the energy evolution of voiced segments. This evolution can be extrapolated to some extent by using the pitch excitation gain values of each subframe of the last good frame. In general, if these gains are greater than 1, the signal energy is increasing, if they are lower than 1, the energy is decreasing. α is thus multiplied by a correction factor f_b computed as follows:

$$f_b = \sqrt{0.1b(0) + 0.2b(1) + 0.3b(2) + 0.4b(3)} \tag{6.5.1-1}$$

where $b(0)$, b (1), b (2) and b (3) are the pitch gains of the four subframes of the last correctly received frame. The value of f_b is constrained between 0.98 and 0.85 before being used to scale the periodic part of the excitation. In this way, strong energy increases and decreases are avoided. For erased frames following a correctly received frame other than UNVOICED, the excitation buffer is updated with this periodic part of the excitation only. This update will be used to construct the adaptive codebook excitation in the next frame.

6.5.2 Construction of the Random Part of the Excitation

The innovative (non-periodic) part of the excitation is generated randomly. A simple random generator with approximately uniform distribution is used. Before adjusting the innovation gain, the randomly generated innovation is scaled to some reference value, fixed here to the unitary energy per sample. At the beginning of an erased block, the innovation gain g_s is initialized by using the innovative excitation gains of each subframe of the last good frame

$$g_s = 0.1g(0) + 0.2g(1) + 0.3g(2) + 0.4g(3) \qquad (6.5.2\text{-}1)$$

where $g(0)$, $g(1)$, $g(2)$ and $g(3)$ are the fixed-codebook, or innovation, gains of the four subframes of the last correctly received frame. The attenuation strategy of the random part of the excitation is somewhat different from the attenuation of the pitch excitation. The reason is that the pitch excitation (and thus the excitation periodicity) is converging to 0 while the random excitation is converging to the CNG excitation energy. The innovation gain attenuation is done as

$$g_s^1 = \alpha g_s^0 + (1-\alpha)g_n \qquad (6.5.2\text{-}2)$$

where g_s^1 is the innovative gain at the beginning of the next frame, g_s^0 is the innovative gain at the beginning of the current frame, g_n is the gain of the excitation used during the comfort noise generation and α is as defined in Table 6.5-1. Similarly to the periodic excitation attenuation, the gain is thus attenuated linearly throughout the frame on a sample-by-sample basis starting with g_s^0 and going to the value of g_s^1 that would be achieved at the beginning of the next frame.

Finally, if the last correctly received frame is different from UNVOICED, the innovation excitation is filtered through a linear phase FIR high-pass filter with coefficients -0.0125, -0.109, 0.7813, -0.109, and -0.0125. To decrease the amount of noisy components during voiced segments, these filter coefficients are multiplied by an adaptive factor equal to (0.75 – 0.25 r_v), with r_v denoting the voicing factor as defined in Equation (6.2.2-2). The random part of the excitation is then added to the adaptive excitation to form the total excitation signal. If the last good frame is UNVOICED, only the innovative excitation is used and it is further attenuated by a factor of 0.8. In this case, the past excitation buffer is updated with the innovation excitation, as no periodic part of the excitation is available.

6.5.3 Spectral Envelope Concealment, Synthesis and Updates

To synthesize the decoded speech, the LP filter parameters must be obtained. The spectral envelope is gradually moved to the estimated envelope of the background noise. Here the ISF representation of LP parameters is used

$$I^1(j) = \alpha I^0(j) + (1-\alpha)I_n(j), \quad j = 0, \ldots, 15 \qquad (6.5.3\text{-}1)$$

In Equation (6.5.3-1), $I^1(j)$ is the value of the jth ISF of the current frame, $I^0(j)$ is the value of the jth ISF of the previous frame, and $I_n(j)$ is the value of the jth ISF of the estimated comfort noise envelope .

The synthesized speech is obtained by filtering the excitation signal through the LP synthesis filter and post-processed similar to the procedure described in Sections 6.3 and 6.4.The filter coefficients are computed from the ISF representation and are interpolated for each subframe (four times per frame) as during normal decoder operation. As innovative gain quantizer and ISF quantizer both use a prediction, their memory will not be up to date after the normal operation is resumed. To reduce this effect, the quantizers' memories are estimated and updated at the end of each erased frame.

6.5.4 Recovery of Normal Operation after an Erasure

The problem of the recovery after an erased block of speech frames is essentially due to the strong prediction. The CELP type speech coders achieve their high signal to noise ratio for voiced speech due to the fact that they exploit the past excitation signal to encode the present frame excitation (long-term or pitch prediction). Also, most of the quantizers (LP quantizers, gain quantizers) make use of a prediction.

6.5.4.1 Artificial Onset Reconstruction

Routine Name: `voiced_onset`

Inputs:

- $u(n)$: The excitation signal from the previous frame
- $h(i)$: The LP synthesis filter impulse response
- E: The synthesized speech energy
- τ_q: First glottal pulse position with respect to frame beginning
- $p(i)$: Pitch lags for each subframe
- $g(i)$: The algebraic codebook gains for each subframe

Outputs:

- $u'(n)$: The excitation signal

Initialization:

- None

The most complicated case related to the use of the long-term prediction is when a voiced onset is lost. The lost onset means that the voiced speech onset happened at some point in the erased block. In this case, the last correctly received frame was unvoiced or inactive and thus no periodic excitation is found in the excitation buffer. The first good speech frame after the erased block is however voiced, the excitation buffer at the encoder is highly periodic and the adaptive excitation has been encoded using this periodic past excitation. As this periodic part of the excitation is completely missing at the decoder, it can take up to several frames to recover from this loss.

If an ONSET class is lost (i.e., a good VOICED frame is received after an erasure, but the last good frame before the erasure was UNVOICED) and the first good frame after the erasure is of Generic FR type, a special technique is used to artificially reconstruct the lost onset and to trigger the voiced synthesis. The technique is used only in Generic FR encoding type as this is the only encoding scheme where all the supplementary information for better frame erasure concealment is transmitted. At the beginning of the 1^{st} good frame after a lost onset, the periodic part of the excitation is constructed artificially as a low-pass filtered periodic train of pulses separated by a pitch period. In this case, the low-pass filter is a simple linear phase FIR filter with the impulse response $\mathrm{h}_{\mathrm{low}} = \{-0.0125, 0.109, 0.7813, 0.109, \text{and} -0.0125\}$. The innovative part of the excitation is constructed using normal CELP decoding.

In practice, the length of the artificial onset is limited so that at least one complete pitch period is constructed by this method and the method is continued to the end of the current subframe. After that, a normal ACELP processing is resumed. The pitch value considered is the rounded average of the decoded pitch values of all subframes where the artificial onset reconstruction is used. The low-pass filtered impulse train is realized by placing the impulse responses of the low-pass filter in the adaptive excitation buffer (previously initialized to zero). The first impulse response will be centered at the quantized position τ_q (transmitted within the bitstream) with respect to the frame beginning and the remaining impulses will be placed with the distance of the averaged pitch up to the end of the last subframe affected by the artificial onset reconstruction.

As an example, let us assume that the pitch values in the first and the second subframe be $p(0)$=70.75 and $p(1)$=71. Since this is larger than the subframe size of 64, then the artificial onset will be constructed during the first two subframes and the pitch period will be equal to the pitch average of the two subframes rounded to the nearest integer; i.e. 71. The last two subframes will be processed by normal ACELP decoder.

The energy of the periodic part of the artificial onset excitation is then scaled by the gain corresponding to the quantized and transmitted energy for FER concealment (as defined in Equations (5.22.2.1-1) and (5.22.2.1-2)) and divided by the gain of the LP synthesis filter. The LP synthesis filter gain is computed as

$$g_{LP} = \sqrt{\sum_{i=0}^{63} h_{LP}^2(i)} \qquad (6.5.4.1\text{-}1)$$

where $h_{LP}(i)$ is the LP synthesis filter impulse response. Finally, the artificial onset gain is reduced by multiplying the periodic part with 0.96. The LP filter for the output speech synthesis is not interpolated in the case of an artificial onset construction. Instead, the received LP parameters are used for the synthesis of the whole frame.

6.5.5 Energy Control

<u>Routine Name</u>: `scale_syn`

Inputs:

- $\hat{s}(n)$: The speech synthesis at 12.8 kHz sampling rate
- $u'(n)$: The excitation signal
- $u(n)$: The enhanced excitation signal
- $\hat{a}_i$: The quantized LP filter coefficients
- $p(i)$: Pitch lags for each subframe
- E: The synthesized speech energy

Outputs:

- $u'(n)$: The excitation signal with controlled energy
- $u(n)$: The enhanced excitation signal with controlled energy

Initialization:

- Buffers, filter memories, and static variables are set to zero at initialization.

The most important task in the recovery after an erased block of speech frames is to properly control the energy of the synthesized signal. The synthesis energy control is needed because of the strong prediction. The energy control is most important when a block of erased frames happens during a voiced segment. When a frame erasure occurs after a voiced frame, the excitation of the last good frame is typically used during the concealment with some attenuation strategy. When a new LP filter occurs with the first good frame after the erasure, there can be a mismatch between the excitation energy and the gain of the new LP synthesis filter. The new synthesis filter can produce a synthesis signal with energy highly different from the energy of the last synthesized erased frame and also from the original signal energy.

The energy control is performed in all active speech frames following an erasure despite the fact that only in Generic FR all the supplementary information necessary for better erasure concealment is transmitted. In Signaling HR encoding scheme, the class information and the energy information is also preserved. In addition to these encoding schemes, the frame class is implicitly transmitted also for VOICED HR frame and UNVOICED HR and QR frames. For the remaining active speech frames, the class is estimated using the measure of signal periodicity r_v (6.2.2-2) averaged over all subframes. The classification is done following the rules in Table 6.5-2.

Table 6.5-2: Signal Classification Rules at the Decoder if not transmitted

Previous Frame Class	Rule	Current Frame Class
ONSET VOICED VOICED TRANSITION	$r_v > -0.1$	VOICED
	$-0.1 \geq r_v \geq -0.5$	VOICED TRANSITION
	$r_v < -0.5$	UNVOICED
UNVOICED TRANSITION UNVOICED	$r_v > -0.1$	ONSET
	$-0.1 \geq r_v \geq -0.5$	UNVOICED TRANSITION
	$r_v < -0.5$	UNVOICED

The energy control during the first good frame after an erased frame can be summarized as follows. The synthesized signal is scaled so that its energy is similar to the energy of the end of the last erased frame at the beginning of the frame and is converging to the transmitted energy towards the end of the frame with preventing a too important energy increase.

The energy control is done in the synthesized speech signal domain. Even if the energy is controlled in the speech domain, the excitation signal must be scaled as it serves as long-term prediction memory for the following frames. The synthesis is then repeated to smooth the transitions. Let g_0 denote the gain used to scale the 1st sample in the current frame and g_1 the gain used at the end of the frame. The excitation signal is then scaled as follows

$$u_s(i) = g_{AGC}(i)u(i)\,, \qquad i=0, \ldots, L-1 \tag{6.5.5-1}$$

where $u_s(i)$ is the scaled excitation, $u(i)$ is the excitation before the scaling, L is the frame length and $g_{AGC}(i)$ is the gain starting from g_0 and converging exponentially to g_1

$$g_{AGC}(i) = f_{AGC}\; g_{AGC}(i-1) + (1 - f_{AGC})\, g_1 \qquad i=0, \ldots, L-1 \tag{6.5.5-2}$$

with the initialization of $g_{AGC}(-1) = g_0$, where f_{AGC} is the attenuation factor set in this implementation to the value of 0.98. This value has been found experimentally as a compromise of having a smooth transition from the previous (erased) frame on one side, and scaling the last pitch period of the current frame as much as possible to the correct (transmitted) value on the other side. This is important because the transmitted energy value is estimated pitch synchronously at the end of the frame. The gains g_0 and g_1 are defined as

$$g_0 = \sqrt{\frac{E_{-1}}{E_0}} \tag{6.5.5-3}$$

$$g_1 = \sqrt{\frac{E_q}{E_1}} \tag{6.5.5-4}$$

where E_{-1} is the energy computed at the end of the previous (erased) frame, E_0 is the energy at the beginning of the current (recovered) frame, E_1 is the energy at the end of the current frame and E_q is the quantized transmitted energy at the end of the current frame, computed at the encoder from Equations (5.22.2.1-1) and (5.22.2.1-2). If E_q is not available, E_q is set to E_1. E_{-1} and E_1 are computed similarly using the synthesized speech signal $\hat{s}(n)$. When E_{-1} is computed pitch synchronously, it uses the concealment pitch period T_c and E_1 uses the last subframe-rounded pitch T_3. E_0 is computed similarly using the rounded pitch value T_0 of the first subframe, the equations (5.22.2.1-1) and (5.22.2.1-2) being modified to

$$E = \max(\hat{s}^2(i)) \qquad i = 0,\ldots,t_E \tag{6.5.5-5}$$

for VOICED and ONSET frames. t_E equals to the rounded pitch lag or twice that length if the pitch is shorter than 64 samples. For other frames,

$$E = \frac{1}{t_E}\sum_{i=0}^{t_E}(\hat{s}^2(i)) \tag{6.5.5-6}$$

with t_E equal to the half of the frame length. The gains g_0 and g_1 are further limited to a maximum allowed value, to prevent strong energy. This value has been set to 1.2 with the exception of Signaling HR frames or very low energy frames (E_q < 1.1). In these two cases, g_1 is limited to 1. If the erasure occurs during a voiced speech segment (i.e. the last good frame before the erasure and the first good frame after the erasure are classified as VOICED TRANSITION, VOICED or ONSET) and E_q is not transmitted, further precautions must be taken because of the possible mismatch between the excitation signal energy and the LP filter gain, mentioned previously. When the LP filter gain of the first frame after an erasure is higher than that of the LP gain of the last frame before the erasure, the energy of the excitation is adjusted to the gain of the new LP filter

$$E_q = E_1 \frac{E_{LP0}}{E_{LP1}} \tag{6.5.5-7}$$

where E_{LP0} is the energy of the LP filter impulse response of the last good frame before the erasure and E_{LP1} is the energy of the LP filter of the first good frame after the erasure. The LP filters of the last subframes of uncorrupted voiced-type frames (VOICED TRANSITION, VOICED or ONSET) are used. Finally, the value of E_q is limited to the value of E_{-1} in this case (voiced segment erasure without E_q information being transmitted).

The following exceptions, all related to transitions in speech signal, further overwrite the computation of g_0. If artificial onset is used in the current frame, g_0 is set to 0.5 g_1, to make the onset energy increase gradually. In the case of a first good frame after an erasure classified as ONSET, the gain g_0 is prevented to be higher than g_1. This precaution is taken to prevent a positive gain adjustment at the beginning of the frame (which is probably still at least partially unvoiced) from amplifying the voiced onset (at the end of the frame). Finally, during a transition from voiced to unvoiced (i.e. that last good frame being classified as VOICED TRANSITION, VOICED or ONSET and the current frame being classified UNVOICED) or during a transition from a non-active speech period to active speech period (last correctly received frame being encoded as comfort noise, current frame being encoded as active speech, but the lost voiced onset not being detected), the value of g_0 is set to g_1.

The synthesized speech is obtained by filtering the excitation signal through the LP synthesis filter and post-processed similar to the procedure described in Sections 6.3 and 6.4.

6.6 Decoding of Inactive Speech Frames (CNG-ER and CNG-QR)

Routine Name: `CNG_dec, CNG_synthesis`

Inputs:

- Decoded quantization indices

Outputs:

- $u'(n)$: The excitation signal

Initialization:

- Initialization is done the same way as described in Section 5.23.

The quantized value of the energy per sample information in the $\log_2$ domain is found by

$$\hat{E}_{s,2} = \frac{24}{63} index - 2 \tag{6.6-1}$$

The quantized energy per sample in the linear domain is then found as

$$\hat{E}_s = 2^{\hat{E}_{s,2}} \tag{6.6-2}$$

The received ISP indices are used to obtain the quantized ISP parameter vector. The initialization and update of the smoothed energy and the LP filter are performed as described in Section 5.23. Also, generating and scaling of the random excitation is performed similar to Section 5.23. Furthermore, memory update is similar to Section 5.21. Note that in case of CNG-ER and CNG-QR frames, the procedures described in Sections 6.1, 6.2, 6.4, and 6.5 are not performed. The synthesized speech is obtained by filtering the excitation signal through the LP synthesis filter and post-processed the same way as described in Section 6.3.

6.7 Detection and Concealment of Frames with Corrupted Rate Information

The detection and concealment of frames with corrupted rate information in the VMR-WB decoder is described in this section. Note that this procedure is not performed in the MIME storage format. The algorithm comprises a number of tests that are performed in two stages. The first stage is performed by default, whereas the second stage is performed only if compilation option BRH_LEVEL2 in VMR-WB C simulation is selected.

6.7.1 Test of Frame Structure

The first level in the bad rate determination algorithm consists of checking the structure of the received frames. The permissible bit combinations for each encoding type are defined in Section 8. All bits in the frame are verified by the decoder and any undefined combination is considered as an indication of a frame with corrupted rate information. Such a frame is subsequently declared and processed as an erasure.

For example, the bit allocation in some encoding types contains some unused bits that are always set to 0 by the encoder. Whenever one of these unused bits is found to be equal to 1, the frame is considered as a bad frame and is processed as a frame erasure. The decoder also checks for some invalid bit combinations.

6.7.2 CNG Frames

The following bad rate determination and concealment procedure applies only to CNG-ER encoding type. For CNG-QR encoding type, on the other hand, the test of frame structure is sufficient to detect most bad rate frames and no further verification is performed.

Bad rate determination for CNG-ER frames is to verify that a received frame labeled as an eighth rate frame is indeed a CNG-ER frame and it is further based on the ISF ordering and on the variation of the quantized energy per sample. The determination and handling procedures are described in the following two sections.

6.7.2.1 Test of the ISF Ordering

A basic property of the ISF parameters is that they are naturally ordered, that is *ISF[0] < ISF[1] <… < ISF[14]*. This property can be used to check whether the decoded ISF vector corresponds to a valid predictor.

1. Detection Method: The frame is considered a bad rate frame and consequently declared as an erasure when the following condition on the decoded ISFs is not verified:

$$ISF[n] > ISF[n-1]-85.0, \text{ for } n=1 \text{ to } 14 \qquad (6.7.2.1\text{-}1)$$

2. Concealment Method: When a frame is declared as a bad rate frame by the above procedure, it is processed as an erasure. Note that the bad rate determination and concealment procedure based on the ISF ordering for CNG-ER frames is subject to the selection of the compilation option BRH_LEVEL2 in VMR-WB C simulation.

6.7.2.2 Test of the Variation of the Quantized Energy

For CNG-ER frames with correct ISF ordering, a second verification is done on the variation of the quantized energy per sample in the linear domain $\hat{E}_s$ (Equation 6.6-2). During the verification process, the CNG parameters (energy per sample and ISP vector) are already decoded and their old values (last correctly received CNG parameters) are also temporarily stored.

1. Detection Method: The received frame is considered as a bad rate frame when

$$\hat{E}_s > 60.0 * \hat{E}_s^{old}, \qquad (6.7.2.2\text{-}1)$$

   where $\hat{E}_s^{old}$ is the old quantized energy per sample in the linear domain.

2. Concealment Method: When a frame is declared as a bad rate frame by the above procedure, the decoded CNG parameters (energy per sample and ISP vector) are replaced by their old value, and the frame is processed as a normal CNG frame.

The above procedure is not performed in the following cases:

- For the first CNG frame received by the decoder (initialization phase)
- When the previous frame was already a CNG frame (regardless of its bit rate)
- When the previous frame was already classified as a bad rate frame in order to prevent consecutive false alarms.

6.7.3 Active Speech Frames

The bad rate determination is performed only on uncorrupted speech frames (i.e., not during frame erasures). There are three methods of bad rate determination for speech frames:

1. Detection based on the ISF ordering
2. Detection by testing the LP gain against the fixed-codebook gain
3. Detection by testing the synthesis energy against the transmitted FER energy information

The detailed bad rate determination procedures for those three methods and the corresponding bad rate concealment procedures are described in the following three sections.

6.7.3.1 Test of the ISF Ordering

1. Detection Method: The frame is considered as a bad rate frame and consequently declared as an erasure when the following condition on the decoded ISFs is not satisfied:

$$\mathrm{ISF}[n] > \mathrm{ISF}[n\text{-}1]\text{-}60.0, \text{ for } n\text{=}1 \text{ to } 14 \qquad (6.7.3.1\text{-}1)$$

2. Concealment Method: When a frame is declared as a bad rate frame by the above procedure, it is processed as an erasure. Note that the bad rate determination and concealment procedure based on the ISF ordering for active speech frames is subject to the selection of compilation option BRH_LEVEL2 in the VMR-WB C simulation.

6.7.3.2 Test of the LP Gain against the Fixed-Codebook Gain

This bad rate determination and concealment procedure is performed once per subframe during the decoding process after the decoding of the gains but before the computation of the global excitation. The following procedure is performed if at least 30 good (i.e., not erased) voiced speech frames have been received since this mechanism has detected the last bad rate frame.

This mechanism is based on the following two parameters:

1. The energy E_{lp} of the impulse response of the interpolated LP speech synthesis filter with quantized coefficients $H(z)$ for the current subframe.
2. The gain of the fixed codebook $\hat{g}_c$ normalized by the energy of the fixed-codebook and by the square root of a long-term estimation of the energy $Ener_{lp}$.

$Ener_{lp}$ is measured on voiced speech frames only and is relative to the nominal level of -26 dBov. It is initialized with a value of 0.1 corresponding to –16 dBov. It is updated at the end of each good voiced speech frame (i.e., it is not updated during CNG, bad or erased frames nor during unvoiced speech frames) using the following formula:

$$Ener_{lp} = 0.95 * OldEner_{lp} + 0.05 * FrameEner \qquad (6.7.3.2\text{-}1)$$

where *FrameEner* is the mean energy per sample of the synthesis speech signal relative to the nominal level of -26 dBov (i.e., the sum of the squares of the reconstructed speech signal samples over the frame length, divided by the frame length and by the square of 1642.0). The value of $Ener_{lp}$ is limited to be always above 0.1 corresponding to -36 dBov.

1. Detection Method: When the LPC gain E_{lp} is above 36, the frame is declared as a bad rate frame. Otherwise, the normalized fixed codebook gain is compared to a detection threshold taken from Table 6.7-2 depending on the encoding type as shown in Table 6.7-1. When the normalized fixed-codebook gain is above that threshold, the frame is declared as a bad rate frame.

2. Concealment Method: When a frame is declared as a bad rate frame, the following modifications are applied to the decoded parameters. First, the decoded fixed-codebook gain is multiplied by 0.1. For unvoiced encoding types, the resulting gain is also limited to a maximum value of 2500. For all other encoding types, the decoded pitch gain is multiplied by 0.25. Finally, the coefficients of the interpolated predictor A(z) are replaced by the coefficients of $A'(z) = A(z/\gamma)$ with γ=0.80.

Note that the following subframes of the current frame are also considered as bad rate frames, and that the same modifications are applied to the decoded parameters.

Table 6.7-1: Threshold selection as a function of the encoding type for the Bad Rate Determination algorithm

Encoding Type	Threshold
Voiced HR	T_{Voiced}
Unvoiced QR, Unvoiced HR	$T_{Unvoiced}$
All other encoding types	T_{Other}

Table 6.7-2: Detection thresholds for the ratio between the LPC gain and the normalized fixed-codebook gain

LPC Gain	T_{Voiced}	$T_{Unvoiced}$	T_{Other}
[0,2]	1594.633545	1077.620972	6929.179688
[2,4]	2012.288574	1603.834839	6918.305176
[4,6]	2012.288574	1719.556519	6915.586426
[6,8]	1962.630127	1883.688232	5165.335938
[8,10]	2608.796387	1883.688232	3597.000977
[10,12]	1842.001221	1883.687744	3198.981934
[12,14]	1262.718506	1459.894287	2306.064453
[14,16]	1078.688721	953.719238	2026.872803
[16,18]	651.626831	827.175537	1165.432739
[18,20]	445.751007	770.602051	942.135376
[20,22]	314.417175	935.903442	899.617188
[22,24]	281.579102	935.903442	1076.359131
[24,26]	198.705811	707.029358	848.216797
[26,28]	137.482468	488.619415	340.947937
[28,30]	96.404831	227.806274	168.581619
[30,32]	68.919113	154.388351	106.822678
[32,34]	62.047684	57.347088	71.641693
[34,36]	45.933205	33.086773	50.357224
[36,38]	39.077591	27.021694	34.045952

6.7.3.3 Test of the Synthesis Energy against the FER Energy Information

In the Generic FR, 14 bits are used to send supplementary information that improves frame erasure concealment and the convergence and recovery of the decoder after erased frames (see Section 5.22). These parameters include, among others, the quantized energy of the synthesis signal E_q and signal classification information. These parameters are also used to detect bad Generic FR frames.

This operation is performed only on Generic FR frames (for which the quantized synthesis energy and the information on signal classification is transmitted as supplementary information). It is not performed when the previous frame was an erasure.

The energy E_s of the decoded synthesized signal is computed after the synthesis filtering operation. As it was mentioned in Section 5.22.2.1, the way the energy E is computed at the encoder depends on the signal classification information. The energy E_s is computed the same way using the decoded classification information.

1. Detection Method: The frame is declared as a bad rate frame when the square root of the ratio between the quantized energy E_q and the energy of the synthesis signal E_s is less than 0.1:

$$\sqrt{\frac{E_q}{E_s}} \leq 0.1 \qquad (6.7.3.3\text{-}1)$$

2. Concealment Method: When a frame is declared as a bad rate frame by the above procedure, the synthesized speech signal is attenuated by a factor of 0.1 using the energy control procedure described in Section 6.5.5. The following gains are used for the energy control:

   $$g_0 = g_1 = 0.1. \qquad (6.7.3.3\text{-}2)$$

The second synthesis operation, also described in Section 6.5.5, is done with modified linear predictors. For each subframe, the coefficients of the interpolated predictors $A(z)$ are replaced by the coefficients of $A'(z) = A(z/\gamma)$ with γ=0.80.

Note that this procedure not only detects bad rate frames when they occur. It can also detect previously overlooked bad rate frames. This is particularly important when an undetected bad rate frame causes a high-energy error. In that case, the bad rate determination procedure based on the synthesis energy can detect the mismatch between the energy of the synthesized signal E_s and the quantized energy E_q in the frames following the overlooked bad rate frame. The corresponding bad rate handling procedure thus limits error propagation and improves the convergence of the decoder.

7 INTEROPERABLE INTERCONNECTION BETWEEN VMR-WB AND AMR-WB CODECS

The AMR-WB interoperable mode of VMR-WB enables encoding and decoding compatibility between VMR-WB and AMR-WB (G.722.2). However, due to differences between the frame formats and other system specific requirements and dependencies in VMR-WB and AMR-WB codecs, interworking functions that reside in an intermediate gateway are required between the two codecs to enable bi-directional interoperability. Note that the interworking functions operate at bit-stream level and have minimal non-computational logic and are much simpler and more efficient than transcoding between the two codecs. Also, they do not result in speech quality degradation under nominal conditions.

In the following sections the forward link interworking function (i.e., AMR-WB to VMR-WB) and reverse-link interworking function (i.e., VMR-WB to AMR-WB) will be described. These interworking functions will ensure compliance with the requirements of the native systems is maintained throughout an interoperable interconnection.

7.1 VMR-WB to AMR-WB Interconnection (Reverse Link)

Routine Name: `vmr2amr`
Inputs:

- The speech data packets generated by VMR-WB encoder

Outputs:

- The speech data packet compatible with AMR-WB frame format

Initialization:
- The internal buffers are reset at initialization.

The source-controlled rate operation (i.e., the VAD/DTX/CNG mechanism) in AMR-WB is shown in Figure 7.1-1 and can be described as follows. Upon termination of an active speech interval, seven background noise frames are encoded as speech frames with the VAD flag set to zero (i.e., DTX hangover). A SID_FIRST frame is then transmitted. In the SID_FIRST frame the signal is not encoded and CNG parameters are derived from the DTX hangover (i.e., the previous 7 speech frames) at the decoder. Note that AMR-WB does not use DTX hangover after active speech periods that are shorter than 24 frames in order to reduce the DTX hangover overhead. After an SID_FIRST frame, two frames are sent as NO_DATA frames, followed by a SID_UPDATE frame encoded at 1.75 kbps (i.e., corresponding to 35 bits/frame). After that, seven NO_DATA frames are transmitted followed by a SID_UPDATE frame and so on. This procedure continues until an active speech frame is detected (VAD_flag=1) [16].

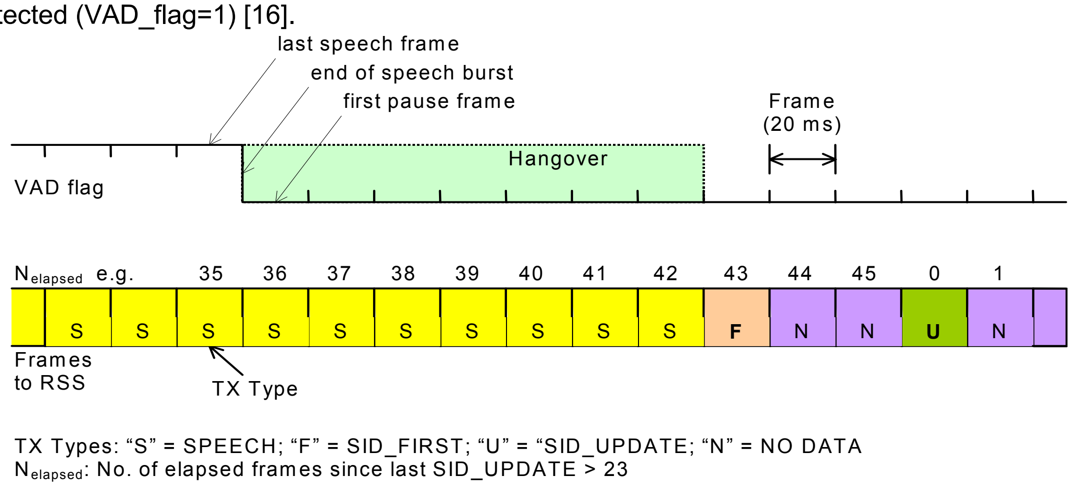

Figure 7.1-1: Normal VAD/DTX/CNG procedure for AMR-WB

The VAD in the VMR-WB codec uses VAD hangover as long as necessary for preserving unvoiced stops and the CNG encoder is used whenever the VAD_flag=0 to encode silence intervals. The VMR-WB encoder does not use DTX hangover, instead, the first background noise frame after an active speech interval is encoded at 1.75 kbps and transmitted using CNG-QR frame type followed by two frames encoded at Eighth-Rate followed by another frame at 1.75 kbps (i.e., CNG-QR). After that, seven frames are transmitted at Eighth-Rate followed by one CNG-QR frame and so on. This procedure roughly simulates the AMR-WB DTX operation with the exception that no DTX hangover is used in order to reduce the ADR of VMR-WB codec and thus reducing the capacity impact of the AMR-WB interoperable mode in the reverse link.

When signaling is requested by the cdma2000 system (e.g., dim and burst), the I-HR encoding type is used. This is to avoid declaring the speech frame as a lost frame. The I-HR encoding type consists of encoding the frame as an Interoperable Full-Rate frame then dropping the extra bits corresponding to the algebraic codebook indices in order to fit the packet size to that of CDMA Rate-Set II Half-Rate (e.g., 142 bits per frame in 12.65 kbps Interoperable Full-Rate). In this case, the bit rate is reduced to the CDMA Rate-Set II Half-Rate frame size. Note that I-HR encoding types (i.e., 12.65, 8.85, and 6.60 kbps) are generated directly on bitstream level from the corresponding I-FR encoding types. Consequently, each I-HR encoding type is used in the same way for dim-and-burst signaling (i.e. when the signaling request is sent to the encoder) and for packet-level signaling (i.e. when Full-Rate frames have to be converted to Half-Rate frames at the base station to accommodate signaling information). Figure 7.1-2 illustrates the techniques that have been used for efficient interoperability between VMR-WB and AMR-WB.

In VMR-WB $\rightarrow$ AMR-WB scenario, the speech frames are encoded in the AMR-WB interoperable mode of the VMR-WB encoder, which uses one of the following possible bit rates: (12.65, 8.85, or 6.60 kbps)/I-FR for active speech frames, (12.65, 8.85, or 6.60 kbps)/I-HR in case of dim-and-burst signaling, CNG-QR to encode silence intervals and background noise frames (one out of eight background noise frame as described above), and CNG-ER frames for most background noise frames (background noise frames not encoded as CNG-QR encoding type). The interworking function shall perform the following procedures:

- Invalid frames are transmitted to the AMR-WB decoder as erased frames (SPEECH_LOST or NO_DATA frame).

- I-FR encoding types are transmitted to AMR-WB decoder as 12.65, 8.85, or 6.60 kbps AMR-WB frames by simply discarding the first byte containing VMR-WB frame identifier. Also the padding bits at the end of the 8.85 and 6.60 I-FR frames shall be discarded. Note that appropriate AMR-WB frame type information is embedded in VMR-WB Interoperable Full-Rate frame structure (see Section 8).

- CNG-QR frames are transmitted to the AMR-WB decoder as SID_UPDATE frames by discarding unused bits at the end.

- CNG-ER frames are transmitted to AMR-WB decoder as NO_DATA frames.

- I-HR encoding types are translated to 12.65, 8.85, or 6.60 kbps frames by generating the missing bits of the algebraic codebook indices. The bits are generated randomly. The first byte containing VMR-WB frame identifier shall also be discarded. The frame identifier bits are used to distinguish different half rate encoding types in the VMR-WB codec. Note that appropriate AMR-WB frame type information is embedded in VMR-WB Interoperable Half-Rate frame structure (see Section 8).

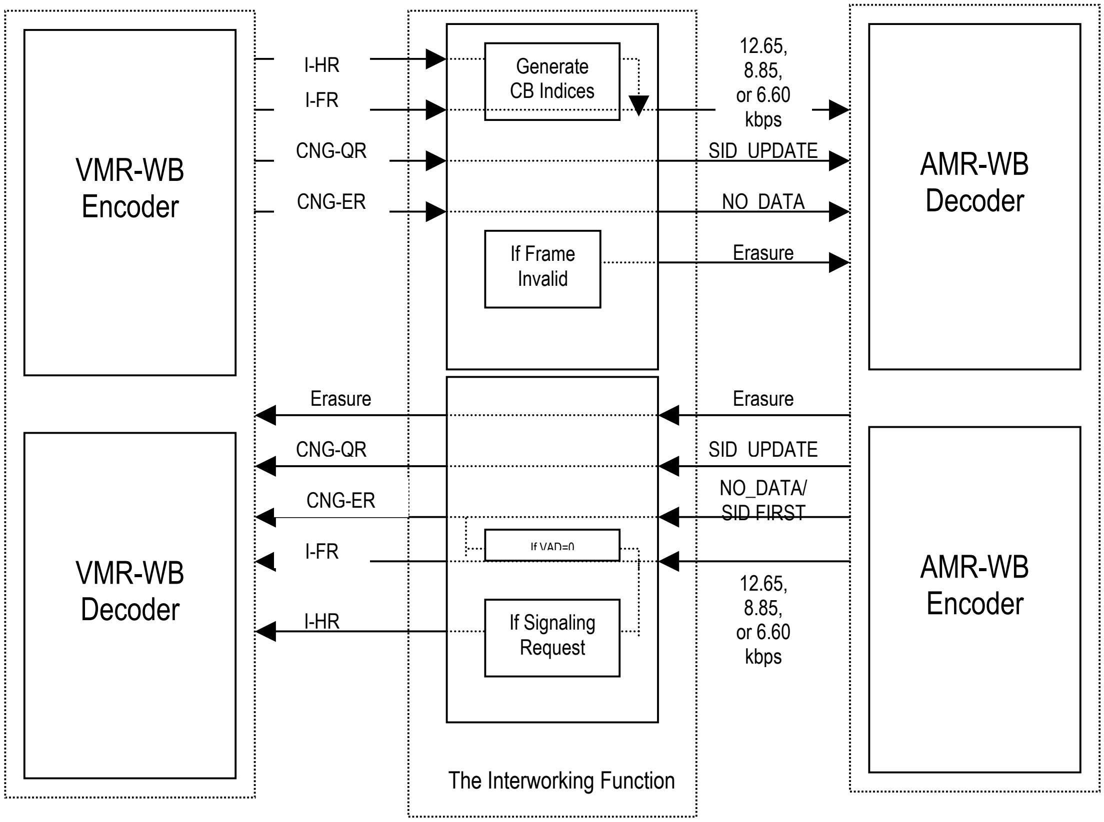

Figure 7.1-2: Interoperable interconnection between AMR-WB and VMR-WB.

7.2 AMR-WB to VMR-WB Interconnection (Forward Link)

Routine Name: amr2`vmr`
Inputs:

- The speech data packets from AMR-WB encoder

Outputs:

- The speech data packet compatible with VMR-WB frame format

Initialization:
- The internal buffers are reset at initialization.

In this scenario, some limitations are imposed by the AMR-WB DTX operation on the interworking function. During active speech, the 1st speech data bit in the incoming AMR-WB bit-stream indicates VAD_flag=0 for DTX hangover period and VAD_flag=1 for active speech. Therefore, the forward link interworking function shall perform the following:

- SID_UPDATE frames are forwarded to VMR-WB as CNG-QR frames.

- SID_FIRST frames and NO_DATA frames are forwarded as CNG-ER frames with a special bit pattern indicating to VMR-WB decoder that the frame corresponds to an AMR-WB NO_DATA frame (see Section 8.4).

- Erased frames (speech lost) are forwarded as ER erasure frames indicated by an illegal bit pattern.

- The first FR frame after active speech with VAD_flag=0 is preserved as FR frame but the following FR frames with VAD_flag=0 are forwarded to VMR-WB as CNG-ER (NO_DATA) frames. This logic is, however, utilized only after at least one valid AMR-WB NO_DATA, SID_FIRST or SID_UPDATE frame has been received from AMR-WB. In this way frame blanking is avoided if AMR WB is not operating in DTX mode.

- If the interworking function is requested to perform signaling operation (e.g., dim and burst signaling) while receiving FR frames from AMR-WB encoder, then the frame is converted into an I-HR frame. This consists of dropping the bits not fitting into CDMA HR frame corresponding to algebraic codebook indices and adding a byte (i.e., VMR-WB frame identifier) in the beginning of the bitstream with an appropriate VMR-WB I-HR encoding type identifier (see Figure 7.1-2 and Section 8.2).

In the VMR-WB decoder, when CNG-ER frames with AMR-WB NO_DATA bit pattern are detected, they are processed by the CNG decoder using the last correctly received CNG parameters. The exception is in the case of the first received AMR-WB NO_DATA frame after an active speech interval. Since the first frame with VAD_flag=0 is transmitted as Full-Rate, the parameters from this frame as well as smoothed last CNG parameters are used to initialize CNG operation:

$$\overline{p} = 0.8\overline{p} + 0.2p_{-1}, \qquad (7.2\text{-}1)$$

where $\overline{p}$ denotes the smoothed CNG parameters (i.e., ISFs and energy) and p_{-1} is the corresponding parameter from the previous Full-Rate frame. The energy information of the previous Full-Rate frame is estimated using the memory of the ACELP adaptive codebook. AMR-WB NO_DATA frames are signaled to the VMR-WB decoder by setting the CNG-ER three ISF indices to 0, 0 and 1, respectively.

Note that by default, 12.65 kbps Interoperable Full-Rate encoding type is used by VMR-WB encoder. However, 8.85 or 6.60 kbps modes can be used by AMR-WB in compliance with the GSM link adaptation mechanism that requires the use of lower rates in case of bad channel conditions. In this case, the VMR-WB codec must be able to encode or decode the lower rates of AMR-WB. For this purpose, special bit patterns are reserved in the frame structure of the VMR-WB Interoperable Full-Rate encoding types to recognize the AMR-WB codec modes 12.65, 8.85, and 6.60 kbps from each other. As shown in Section 8.1, the first byte of the VMR-WB Interoperable Full-Rate frame is the same for all three encoding types (i.e., the frame identifier), then speech data bits that have been arranged according to AMR-WB IF2 frame structure will follow. Therefore, it is sufficient to look into AMR-WB frame type (i.e., first 4 bits of the 2nd byte) to differentiate among the three AMR-WB codec modes. Similarly, three Interoperable Half-Rate encoding types are used corresponding to the three Interoperable Full-Rate encoding types by dropping some of the bits corresponding to the algebraic codebook indices. As shown in Section 8.2, the first byte of the frame is the same for all three encoding types and they are differentiated by the AMR-WB frame type information.

It must be noted that mode switching is not allowed when VMR-WB operates in the AMR-WB interoperable mode. However, while in the AMR-WB interoperable mode (i.e., VMR-WB mode 3), the AMR-WB codec mode request can be signaled to the VMR-WB encoder by “-f” option in the C simulation. In this case, VMR-WB encoder can encode input speech using AMR-WB codec modes 0, 1, and 2 corresponding to 6.60, 8.85, and 12.65 kbps, respectively.

8 VMR-WB FRAME STRUCTURE

To further simplify interoperable interconnections between VMR-WB and AMR-WB, the frame structure and bit stream of VMR-WB closely follow that of AMR-WB Interface Format 2 (IF2) [16]. This section describes the frame structure and detailed bit position corresponding to all existing encoding types of VMR-WB. In general, the frame structure of VMR-WB in various encoding schemes comprises a preamble (i.e., frame type identifier) and the speech data bits.

The speech data field in Generic Full-rate, Signaling Half-rate, and in all interoperable encoding types is formatted according to AMR-WB IF2 and follows the prioritization of bits in class A, class B, and class C as described in [16]. In the VMR-WB interoperable encoding types, the AMR-WB Frame Type (FT) field is appropriately set by the VMR-WB encoder according to the AMR-WB codec mode and the Frame Quality Indicator bit (FQI) is always set to 1 by the VMR-WB encoder.

8.1 Frame Structure of Full-Rate Encoding Types

The following sections describe the frame structure and the position of bits in the bit stream for various full-rate encoding schemes in VMR-WB encoder.

8.1.1 Frame Structure of Generic Full-Rate

The mapping of bits for the Generic Full-Rate is shown in Table 8.1-1. To prioritize the speech data bits according to AMR-WB IF2 format, the following mapping function is used:

For j = 1 to K-1

$$D[j-1] = s(table_2[j]+1);$$

Where $table_2[j]$ is given in Table 8.1-3and K-1 denotes the total number of speech data bits produced by VMR-WB codec for this encoding type $(K=253)$ excluding the FER protection bits. The source bits $s(.)$ represent the encoder generated parameters in order of occurrence as shown in Table 8.1-2.

Table 8.1-1: Mapping of bits in VMR-WB Generic Full-Rate

	MSB	**Mapping of Bits in VMR-WB Generic Full-Rate**						**LSB**
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Frame Type Identifier/ Frame Error Protection Bits Any combination of 6 bits other than 111110						Frame Error Protection Bits	
	FER[0]	FER[1]	FER[2]	FER[3]	FER[4]	FER[5]	FER[6]	FER[7]
2	Frame Error Protection Bits						Speech Data	
	FER[8]	FER[9]	FER[10]	FER[11]	FER[12]	FER[13]	D[0]	D[1]
3	Speech Data							
	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]	D[8]	D[9]
4-33	Speech Data							
	D[10]	D[11]	..	..	..	..	D[248]	D[249]
34	Speech Data							
	D[250]	D[251]						

The following table shows the order of the bits produced by the speech encoder prior to reordering according to AMR-WB IF2 format. Note that the most significant bit (MSB) of each codec parameter is always used first.

Table 8.1-2: Source encoder output parameters in order of occurrence

Bits (MSB-LSB)	Description
s1	VAD-flag
s2 – s9	Index of 1st ISF sub-vector
s10 – s17	Index of 2nd ISF sub-vector
s18 - s23	Index of 3rd ISF sub-vector
s24 – s30	Index of 4th ISF sub-vector
s31 – s37	Index of 5th ISF sub-vector
s38 – s42	Index of 6th ISF sub-vector
s43 – s47	Index of 7th ISF sub-vector
Subframe 1	
s48 – s56	Adaptive codebook index
s57	LTP-filtering-flag
s58 – s66	Fixed-Codebook index for track 1
s67 – s75	Fixed-Codebook index for track 2
s76 – s84	Fixed-Codebook index for track 3
s85 – s93	Fixed-Codebook index for track 4
s94 – s100	VQ gain
Subframe 2	
s101 – s106	Adaptive codebook index (relative)
s107 – s150	Same description as s57 – s100
Subframe 3	
s151 – s203	Same description as s48 – s100
Subframe 4	
s204 – s253	Same description as s101 – s150

The following ordering table shall be read from left to right so that the first element (top left corner) of the table has index 0 and the last element (the rightmost element of the last row) has the index *K*-1 where *K* is the total number of speech bits in the specific encoding type. Note that in Generic FR the VAD bit s1 is replaced by FER[13].

Table 8.1-3: Ordering of the speech encoder bits for Generic Full-Rate: $table_2(j)$

0	4	6	93	143	196	246	7	5	3
47	48	49	50	51	150	151	152	153	154
94	144	197	247	99	149	202	252	96	146
199	249	97	147	200	250	100	203	98	148
201	251	95	145	198	248	52	2	1	101
204	155	19	21	12	17	18	20	16	25
13	10	14	24	23	22	26	8	15	53
156	31	102	205	9	33	11	103	206	54
157	28	27	104	207	34	35	29	46	32
30	55	158	37	36	39	38	40	105	208
41	42	43	44	45	56	106	159	209	57
66	75	84	107	116	125	134	160	169	178
187	210	219	228	237	58	108	161	211	62
112	165	215	67	117	170	220	71	121	174
224	76	126	179	229	80	130	183	233	85
135	188	238	89	139	192	242	59	109	162
212	63	113	166	216	68	118	171	221	72
122	175	225	77	127	180	230	81	131	184
234	86	136	189	239	90	140	193	243	60
110	163	213	64	114	167	217	69	119	172
222	73	123	176	226	78	128	181	231	82
132	185	235	87	137	190	240	91	141	194
244	61	111	164	214	65	115	168	218	70
120	173	223	74	124	177	227	79	129	182
232	83	133	186	236	88	138	191	241	92
142	195	245							

8.1.2 Frame Structure of 12.65 kbps Interoperable Full-Rate

The mapping of bits for the 12.65 kbps Interoperable Full-Rate is shown in Table 8.1-4. To prioritize the speech data bits according to AMR-WB IF2 format, the following mapping function is used:

For j = 0 to K-1

$D[j] = s(table_2[j]+1);$

Where $table_2[j]$ is given in Table 8.1-3 and K denotes the total number of speech data bits produced by VMR-WB codec for this encoding type $(K=253)$. The source bits $s(.)$ represent the encoder generated parameters in order of occurrence as shown in Table 8.1-2.

Table 8.1-4: Mapping of bits in VMR-WB 12.65 kbps Interoperable Full-Rate

<table>
<tr><th></th><th>MSB</th><th colspan="6">Mapping of bits in
VMR-WB 12.65 kbps Interoperable Full-Rate</th><th>LSB</th></tr>
<tr><th>Octet</th><th>Bit 8</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th></tr>
<tr><td rowspan="2">1</td><td colspan="8">Frame Type Identifier</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td rowspan="2">2</td><td colspan="4">AMR-WB Frame Type</td><td>FQI</td><td colspan="3">Speech Data</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td><td>D[0]</td><td>D[1]</td><td>D[2]</td></tr>
<tr><td rowspan="2">3</td><td colspan="8">Speech Data</td></tr>
<tr><td>D[3]</td><td>D[4]</td><td>D[5]</td><td>D[6]</td><td>D[7]</td><td>D[8]</td><td>D[9]</td><td>D[10]</td></tr>
<tr><td rowspan="2">4-33</td><td colspan="8">Speech Data</td></tr>
<tr><td>D[11]</td><td>D[12]</td><td>..</td><td>..</td><td>..</td><td>..</td><td>D[249]</td><td>D[250]</td></tr>
<tr><td rowspan="2">34</td><td colspan="2">Speech Data</td></tr>
<tr><td>D[251]</td><td>D[252]</td></tr>
</table>

8.1.3 Frame Structure of 8.85 kbps Interoperable Full-Rate

The mapping of bits for the 8.85 kbps Interoperable Full-Rate is shown in Table 8.1-5. To prioritize the speech data bits according to AMR-WB IF2 format, the following mapping function is used:

For j = 0 to K-1

$$D[j] = s(table_1[j]+1);$$

Where $table_1[j]$ is given in Table 8.1-7 and K denotes the total number of speech data bits produced by VMR-WB codec for this encoding type *(K=177).* The source bits *s(.)* represent the encoder generated parameters in order of occurrence as shown in Table 8.1-6. Padding bits (zeros) are added to the end of the frame to adjust the frame size to 266.

Table 8.1-5: Mapping of bits in VMR-WB 8.85 kbps Interoperable Full-Rate

	MSB	**Mapping of bits in VMR-WB 8.85 kbps Interoperable Full-Rate**						**LSB**
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Frame Type Identifier							
	1	1	1	1	1	0	0	0
2	AMR-WB Frame Type				FQI	Speech Data		
	0	0	0	1	1	D[0]	D[1]	D[2]
3	Speech Data							
	D[3]	D[4]	D[5]	D[6]	D[7]	D[8]	D[9]	D[10]
4-23	Speech Data							
	D[11]	D[12]	..	..	..	..	D[169]	D[170]
24	Speech Data						Padding Bits	
	D[171]	D[172]	D[173]	D[174]	D[175]	D[176]	0	0
25-33	Padding Bits							
	0	0	..	..	..	..	0	0
34	Padding Bits							
	0	0						

The following table shows the order of the bits produced by the speech encoder. Note that the most significant bit (MSB) of each codec parameter is always used first.

Table 8.1-6: Source encoder output parameters in order of occurrence

Bits (MSB-LSB)	Description
s1	VAD-flag
s2 – s9	Index of 1st ISF sub-vector
s10 – s17	Index of 2nd ISF sub-vector
s18 - s23	Index of 3rd ISF sub-vector
s24 – s30	Index of 4th ISF sub-vector
s31 – s37	Index of 5th ISF sub-vector
s38 – s42	Index of 6th ISF sub-vector
s43 – s47	Index of 7th ISF sub-vector
Subframe 1	
s48 – s55	Adaptive codebook index
s56 – s60	Fixed-Codebook index for track 1
s61 – s65	Fixed-Codebook index for track 2
s66 – s70	Fixed-codebook index for track 3
s71 - s75	Fixed-Codebook index for track 4
s76 – s81	VQ gain
Subframe 2	
s82 – s86	Adaptive codebook index (relative)
s87 – s112	Same description as s56 – s81
Subframe 3	
s113 – s146	Same description as s48 – s81
Subframe 4	
s147 – s177	Same description as s82 – s112

The following ordering table shall be read from left to right so that the first element (top left corner) of the table has index 0 and the last element (the rightmost element of the last row) has the index *K*-1 where *K* is the total number of speech bits in the specific encoding type.

Table 8.1-7: Ordering of the speech encoder bits for 8.85 kbps Interoperable Full-Rate: $table_1(j)$

0	4	6	7	5	3	47	48	49	112
113	114	75	106	140	171	80	111	145	176
77	108	142	173	78	109	143	174	79	110
144	175	76	107	141	172	50	115	51	2
1	81	116	146	19	21	12	17	18	20
16	25	13	10	14	24	23	22	26	8
15	52	117	31	82	147	9	33	11	83
148	53	118	28	27	84	149	34	35	29
46	32	30	54	119	37	36	39	38	40
85	150	41	42	43	44	45	55	60	65
70	86	91	96	101	120	125	130	135	151
156	161	166	56	87	121	152	61	92	126
157	66	97	131	162	71	102	136	167	57
88	122	153	62	93	127	158	67	98	132
163	72	103	137	168	58	89	123	154	63
94	128	159	68	99	133	164	73	104	138
169	59	90	124	155	64	95	129	160	69
100	134	165	74	105	139	170			

8.1.4 Frame Structure of 6.60 kbps Interoperable Full-Rate

The mapping of bits for the 6.60 kbps Interoperable Full-Rate is shown in Table 8.1-8. To prioritize the speech data bits according to AMR-WB IF2 format, the following mapping function is used:

For *j* = 0 to *K*-1

$D[j] = s(table_0[j]+1)$;

Where $table_0[j]$ is given in Table 8.1-10 and *K* denotes the total number of speech data bits produced by VMR-WB codec for this encoding type *(K=132).* The source bits *s(.)* represent the encoder generated parameters in order of occurrence as shown in Table 8.1-9. Padding bits (zeros) shall be added to the end of the frame to adjust the frame size to 266.

Table 8.1-8: Mapping of bits in VMR-WB 6.60 kbps Interoperable Full-Rate

	MSB	**Mapping of bits in VMR-WB 6.60 kbps Interoperable Full-Rate**						**LSB**
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Frame Type Identifier							
	1	1	1	1	1	0	0	0
2	AMR-WB Frame Type				FQI	Speech Data		
	0	0	0	0	1	D[0]	D[1]	D[2]
3	Speech Data							
	D[3]	D[4]	D[5]	D[6]	D[7]	D[8]	D[9]	D[10]
4-18	Speech Data							
	D[11]	D[12]	..	..	..	..	D[129]	D[130]
19	Speech Data	Padding Bits						
	D[131]	0	0	0	0	0	0	0
20-33	Padding Bits							
	0	0	..	..	..	..	0	0
34	Padding Bits							
	0	0						

The following table shows the order of the bits produced by the speech encoder. Note that the most significant bit (MSB) of each codec parameter is always used first.

Table 8.1-9: Source encoder output parameters in order of occurrence

Bits (MSB-LSB)	Description
s1	VAD-flag
s2 – s9	Index of 1st ISF sub-vector
s10 – s17	Index of 2nd ISF sub-vector
s18 – s24	Index of 3rd ISF sub-vector
s25 – s31	Index of 4th ISF sub-vector
s32 – s37	Index of 5th ISF sub-vector
Subframe 1	
s38 – s45	Adaptive codebook index
s46 - 57	Fixed-Codebook Index
s58 – s63	VQ gain
Subframe 2	
s64 – s68	Adaptive codebook index (relative)
s69 – s86	Same description as s46 – s63
Subframe 3	
s87 – s109	Same description as s64 – s86
Subframe 4	
s110 – s132	Same description as s64 – s86

The following ordering table shall be read from left to right so that the first element (top left corner) of the table has index 0 and the last element (the rightmost element of the last row) has the index *K*-1 where *K* is the total number of speech bits in the specific encoding type.

Table 8.1-10: Ordering of the speech encoder bits for 6.60 kbps Interoperable Full-Rate: $table_0(j)$

0	5	6	7	61	84	107	130	62	85
8	4	37	38	39	40	58	81	104	127
60	83	106	129	108	131	128	41	42	80
126	1	3	57	103	82	105	59	2	63
109	110	86	19	22	23	64	87	18	20
21	17	13	88	43	89	65	111	14	24
25	26	27	28	15	16	44	90	66	112
9	11	10	12	67	113	29	30	31	32
34	33	35	36	45	51	68	74	91	97
114	120	46	69	92	115	52	75	98	121
47	70	93	116	53	76	99	122	48	71
94	117	54	77	100	123	49	72	95	118
55	78	101	124	50	73	96	119	56	79
102	125								

8.2 Frame Structure of Half-Rate Encoding Types

The following sections describe the frame structure and the position of bits in the bit stream for various half-rate encoding schemes in VMR-WB encoder.

8.2.1 Frame Structure of Generic Half-Rate

The mapping of bits for the Generic Half-Rate is shown in Table 8.2-1.

Table 8.2-1: Mapping of bits in VMR-WB Generic Half-Rate

	MSB	**Mapping of bits in VMR-WB Generic Half-Rate**						**LSB**
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Identifier	ISFs						
	0	ISF[0]	ISF[1]	ISF[2]	ISF[3]	ISF[4]	ISF[5]	ISF[6]
2	ISFs							
	ISF[7]	ISF[8]	ISF[9]	ISF[10]	ISF[11]	ISF[12]	ISF[13]	ISF[14]
3	ISFs							
	ISF[15]	ISF[16]	ISF[17]	ISF[18]	ISF[19]	ISF[20]	ISF[21]	ISF[22]
4	ISFs							
	ISF[23]	ISF[24]	ISF[25]	ISF[26]	ISF[27]	ISF[28]	ISF[29]	ISF[30]
5	ISFs					Pitch Delay for Sub-Frame 1		
	ISF[31]	ISF[32]	ISF[33]	ISF[34]	ISF[35]	P[0]	P[1]	P[2]
6	Pitch Delay for Sub-Frame 1					FCB Indices for Sub-Frame 1		
	P[3]	P[4]	P[5]	P[6]	P[7]	FCB[0]	FCB[1]	FCB[2]
7	FCB Indices for Sub-Frame 1							
	FCB[3]	FCB[4]	FCB[5]	FCB[6]	FCB[7]	FCB[8]	FCB[9]	FCB[10]
8		Gain Flag	FCB Gain for Sub-Frame 1					
	FCB[11]	G_Flag	G[0]	G[1]	G[2]	G[3]	G[4]	G[5]
9	FCB Indices for Sub-Frame 2							
	FCB[0]	FCB[1]	FCB[2]	FCB[3]	FCB[4]	FCB[5]	FCB[6]	FCB[7]
10	FCB Indices for Sub-Frame 2				FCB Gain for Sub-Frame 2			
	FCB[8]	FCB[9]	FCB[10]	FCB[11]	G[0]	G[1]	G[2]	G[3]
11			Pitch Delay for Sub-Frame 3					
	G[4]	G[5]	P[0]	P[1]	P[2]	P[3]	P[4]	FCB[0]

12	FCB Indices for Sub-Frame 3							
	FCB[1]	FCB[2]	FCB[3]	FCB[4]	FCB[5]	FCB[6]	FCB[7]	FCB[8]
13	FCB Indices for Sub-Frame 3			Gain Flag	FCB Gain for Sub-Frame 3			
	FCB[9]	FCB[10]	FCB[11]	G_Flag	G[0]	G[1]	G[2]	G[3]
14			FCB Indices for Sub-Frame 4					
	G[4]	G[5]	FCB[0]	FCB[1]	FCB[2]	FCB[3]	FCB[4]	FCB[5]
15	FCB Indices for Sub-Frame 4							
	FCB[6]	FCB[7]	FCB[8]	FCB[9]	FCB[10]	FCB[11]	G[0]	G[1]
16	FCB Gain for Sub-Frame 4							
	G[2]	G[3]	G[4]	G[5]				

8.2.2 Frame Structure of Signaling Half-Rate

The mapping of bits for the Signaling Half-Rate, shown in Table 8.2-2, is similar to that of Generic Full-Rate with the exception that the fixed-codebook indices exceeding the HR frame size are dropped from the end of the packet at bitstream level. The Signaling Half-Rate frame is generated during transmission from Generic FR frame to accommodate signaling information at the base station and is not generated by the VMR-WB encoder. To prioritize the speech data bits according to AMR-WB IF2 format, the following mapping function is used:

For j = 1 to K-1

$D[j-1] = s(table_2[j]+1);$

Where $table_2[j]$ is given in Table 8.1-3 and K-1 denotes the total number of speech data bits produced by VMR-WB codec for this encoding type excluding the FER protection bits $(K=110)$. The source bits $s(.)$ represent the encoder generated parameters in order of occurrence as shown in Table 8.1-2.

Table 8.2-2: Mapping of bits in VMR-WB Signaling Half-Rate

	MSB	**Mapping of bits in VMR-WB Signaling Half-Rate**						**LSB**
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Frame Type Identifier			Frame Error Protection Bits Any combination of 6 bits other than 111110				
	1	1	0	FER[0]	FER[1]	FER[2]	FER[3]	FER[4]
2	Frame Error Protection Bits Any combination of 6 bits other than 111110	Frame Error Protection Bits		Unused bits				
	FER[5]	FER[6]	FER[7]	0	0	0	D[0]	D[1]
3	Speech Data							
	D[2]	D[3]	D[4]	D[5]	D[6]	D[7]	D[8]	D[9]
4-15	Speech Data							
	D[10]	D[11]	..	..	..	..	D[104]	D[105]
16	Speech Data							
	D[106]	D[107]	D[108]	D[109]				

8.2.3 Frame Structure of Voiced Half-Rate

Table 8.2-3: Mapping of bits in VMR-WB Voiced Half-Rate

	MSB	**Mapping of bits in VMR-WB Voiced Half-Rate**						**LSB**

Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Frame Type Identifier			ISFs				
	1	1	1	ISF[0]	ISF[1]	ISF[2]	ISF[3]	ISF[4]
2	ISFs							
	ISF[5]	ISF[6]	ISF[7]	ISF[8]	ISF[9]	ISF[10]	ISF[11]	ISF[12]
3	ISFs							
	ISF[13]	ISF[14]	ISF[15]	ISF[16]	ISF[17]	ISF[18]	ISF[19]	ISF[20]
4	ISFs							
	ISF[21]	ISF[22]	ISF[23]	ISF[24]	ISF[25]	ISF[26]	ISF[27]	ISF[28]
5	ISFs							
	ISF[29]	ISF[30]	ISF[31]	ISF[32]	ISF[33]	ISF[34]	ISF[35]	P[0]
6	Pitch Delay							
	P[1]	P[2]	P[3]	P[4]	P[5]	P[6]	P[7]	P[8]
7	Pitch Filtering	FCB Indices for Sub-Frame 1						
	PF	FCB[0]	FCB[1]	FCB[2]	FCB[3]	FCB[4]	FCB[5]	FCB[6]
8	FCB Indices for Sub-Frame 1					Gain Flag		
	FCB[7]	FCB[8]	FCB[9]	FCB[10]	FCB[11]	G_Flag	G[0]	G[1]
9	FCB Gain for Sub-Frame 1				FCB Indices for Sub-Frame 2			
	G[2]	G[3]	G[4]	G[5]	FCB[0]	FCB[1]	FCB[2]	FCB[3]
10	FCB Indices for Sub-Frame 2							
	FCB[4]	FCB[5]	FCB[6]	FCB[7]	FCB[8]	FCB[9]	FCB[10]	FCB[11]
11	FCB Gain for Sub-Frame 2						Pitch Filtering	
	G[0]	G[1]	G[2]	G[3]	G[4]	G[5]	PF	FCB[0]
12	FCB Indices for Sub-Frame 3							
	FCB[1]	FCB[2]	FCB[3]	FCB[4]	FCB[5]	FCB[6]	FCB[7]	FCB[8]
13	FCB Indices for Sub-Frame 3			Gain Flag	FCB Gain for Sub-Frame 3			
	FCB[9]	FCB[10]	FCB[11]	G_Flag	G[0]	G[1]	G[2]	G[3]
14			FCB Indices for Sub-Frame 4					
	G[4]	G[5]	FCB[0]	FCB[1]	FCB[2]	FCB[3]	FCB[4]	FCB[5]
15	FCB Indices for Sub-Frame 4							
	FCB[6]	FCB[7]	FCB[8]	FCB[9]	FCB[10]	FCB[11]	G[0]	G[1]
16	FCB Gain for Sub-Frame 4							
	G[2]	G[3]	G[4]	G[5]				

8.2.4 Frame Structure of Unvoiced Half-Rate

The mapping of bits for the Unvoiced Half-Rate encoding type is shown in Table 8.2-4.

Table 8.2-4: Mapping of bits in VMR-WB Unvoiced Half-Rate

	MSB	**Mapping of bits in VMR-WB Unvoiced Half-Rate**						**LSB**
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Frame Identifier		ISFs					
	1	0	ISF[0]	ISF[1]	ISF[2]	ISF[3]	ISF[4]	ISF[5]
2-5	ISFs							
	ISF[6]	ISF[7]	..	..	..	..	ISF[36]	ISF[37]
6	ISFs							
	ISF[38]	ISF[39]	ISF[40]	ISF[41]	ISF[42]	ISF[43]	ISF[44]	ISF[45]
7	FCB Indices for Sub-Frame 1							
	FCB[0]	FCB[1]	FCB[2]	FCB[3]	FCB[4]	FCB[5]	FCB[6]	FCB[7]
8	FCB Indices for Sub-Frame 1					FCB Gain for Sub-Frame 1		
	FCB[8]	FCB[9]	FCB[10]	FCB[11]	FCB[12]	G[0]	G[1]	G[2]
9	FCB Gain for Sub-Frame 1			FCB Indices for Sub-Frame 2				
	G[3]	G[4]	G[5]	FCB[0]	FCB[1]	FCB[2]	FCB[3]	FCB[4]
10	FCB Indices for Sub-Frame 2							
	FCB[5]	FCB[6]	FCB[7]	FCB[8]	FCB[9]	FCB[10]	FCB[11]	FCB[12]
11	FCB Gain for Sub-Frame 2							
	G[0]	G[1]	G[2]	G[3]	G[4]	G[5]	FCB[0]	FCB[1]
12	FCB Indices for Sub-Frame 3							

	FCB[2]	FCB[3]	FCB[4]	FCB[5]	FCB[6]	FCB[7]	FCB[8]	FCB[9]
13	FCB Indices for Sub-Frame 3			FCB Gain for Sub-Frame 3				
	FCB[10]	FCB[11]	FCB[12]	G[0]	G[1]	G[2]	G[3]	G[4]
14		FCB Indices for Sub-Frame 4						
	G[5]	FCB[0]	FCB[1]	FCB[2]	FCB[3]	FCB[4]	FCB[5]	FCB[6]
15	FCB Indices for Sub-Frame 4							
	FCB[7]	FCB[8]	FCB[9]	FCB[10]	FCB[11]	FCB[12]	G[0]	G[1]
16	FCB Gain for Sub-Frame 4							
	G[2]	G[3]	G[4]	G[5]				

8.2.5 Frame Structure of 12.65 kbps Interoperable Half-Rate

The mapping of bits for the 12.65 kbps Interoperable Half-Rate, shown in Table 8.2-5, is similar to that of 12.65 kbps Interoperable Full-Rate with the exception that the fixed-codebook indices exceeding the HR frame size are dropped from the end of the packet. To prioritize the speech data bits according to AMR-WB IF2 format, the following mapping function is used:

For j = 0 to K-1

$D[j] = s(table_2[j]+1);$

where $table_2[j]$ is given in Table 8.1-3 and K denotes the total number of speech data bits produced by the VMR-WB codec for this encoding type *(K=111)*. The source bits *s (.)* represent the encoder generated parameters, excluding those bits that correspond to the fixed-codebook indices exceeding the HR frame size, in order of occurrence as shown in Table 8.1-2.

Table 8.2-5: Mapping of bits in VMR-WB 12.65 kbps Interoperable Half-Rate

	MSB	**Mapping of bits in VMR-WB 12.65 kbps Interoperable Half-Rate**						**LSB**
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Frame Type Identifier							
	1	1	0	1	1	1	1	1
2	AMR-WB Frame Type				FQI	Speech Data		
	0	0	1	0	1	D[0]	D[1]	D[2]
3	Speech Data							
	D[3]	D[4]	D[5]	D[6]	D[7]	D[8]	D[9]	D[10]
4-15	Speech Data							
	D[11]	D[12]	..	..	..	..	D[105]	D[106]
16	Speech Data							
	D[107]	D[108]	D[109]	D[110]				

8.2.6 Frame Structure of 8.85 kbps Interoperable Half-Rate

The mapping of bits for the 8.85 kbps Interoperable Half-Rate is shown in Table 8.2-6. To prioritize the speech data bits according to AMR-WB IF2 format, the following mapping function is used:

For j = 0 to K-1

$D[j] = s(table_1[j]+1);$

where $table_1[j]$ is given in Table 8.1-7 and K denotes the total number of speech data bits produced by the VMR-WB codec for this encoding type *(K=111)*. The source bits *s(.)* represent the encoder generated parameters, excluding those bits that correspond to the fixed-codebook indices exceeding the HR frame size, in order of occurrence as shown in Table 8.1-6.

Table 8.2-6: Mapping of bits in VMR-WB 8.85 kbps Interoperable Half-Rate

	MSB	**Mapping of bits in VMR-WB 8.85 kbps Interoperable Half-Rate**	**LSB**

Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Frame Type Identifier							
	1	1	0	1	1	1	1	1
2	AMR-WB Frame Type				FQI	Speech Data		
	0	0	0	1	1	D[0]	D[1]	D[2]
3	Speech Data							
	D[3]	D[4]	D[5]	D[6]	D[7]	D[8]	D[9]	D[10]
4-13	Speech Data							
	D[11]	D[12]	..	..	..	..	D[89]	D[90]
14	Speech Data							
	D[91]	D[92]	D[93]	D[94]	D[95]	D[96]	D[97]	D[98]
15	Speech Data							
	D[99]	D[100]	..	..	..	..	D[105]	D[106]
16	Speech Data							
	D[107]	D[108]	D[109]	D[110]				

8.2.7 Frame Structure of 6.60 kbps Interoperable Half-Rate

The mapping of bits for the 6.60 kbps Interoperable Half-Rate is shown in Table 8.2-7. To prioritize the speech data bits according to AMR-WB IF2 format, the following mapping function is used:

For j = 0 to K-1

$D[j] = s(table_0[j]+1);$

where $table_0[j]$ is given in Table 8.1-10 and K denotes the total number of speech data bits produced by the VMR-WB codec for this encoding type (K=111). The source bits $s(.)$ represent the encoder generated parameters, excluding those bits that correspond to the fixed-codebook indices exceeding the HR frame size, in order of occurrence as shown in Table 8.1-9.

Table 8.2-7: Mapping of bits in VMR-WB 6.60 kbps Interoperable Half-Rate

	MSB	**Mapping of bits in VMR-WB 6.60 kbps Interoperable Half-Rate**						**LSB**
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Frame Type Identifier							
	1	1	0	1	1	1	1	1
2	AMR-WB Frame Type				FQI	Speech Data		
	0	0	0	0	1	D[0]	D[1]	D[2]
3	Speech Data							
	D[3]	D[4]	D[5]	D[6]	D[7]	D[8]	D[9]	D[10]
4-12	Speech Data							
	D[11]	D[12]	..	..	..	..	D[81]	D[82]
13	Speech Data							
	D[83]	D[84]	D[85]	D[86]	D[87]	D[88]	D[89]	D[90]
14-15	Speech Data							
	D[91]	D[92]	..	..	..	..	D[105]	D[106]
16	Speech Data							
	D[107]	D[108]	D[109]	D[110]				

8.3 Frame Structure of Quarter-Rate Encoding Types

The following sections describe the frame structure and the position of bits in the bit stream for various quarter-rate encoding schemes in VMR-WB encoder.

8.3.1 Frame Structure of CNG Quarter-Rate

The mapping of the bits in the VMR-WB CNG Quarter-Rate is similar to that of AMR-WB SID_UPDATE [16] and it is shown in Table 8.3-1.

The AMR-WB mode indication bits denoted by xx in the following Table are set to ‘00’, ‘01’, or ‘10’, if VMR-WB generates CNG frames for AMR-WB at 6.60, 8.85, or 12.65 kbps, respectively. By default, these bits are set to ‘10’.

Table 8.3-1: Mapping of bits in VMR-WB CNG Quarter-Rate

	MSB	**Mapping of bits VMR-WB CNG Quarter-Rate**						**LSB**
Octet	**Bit 8**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**
1	Frame Type Identifier				FQI	ISFs		
	1	0	0	1	1	ISF[0]	ISF[1]	ISF[2]
2	ISFs							
	ISF[3]	ISF[4]	ISF[5]	ISF[6]	ISF[7]	ISF[8]	ISF[9]	ISF[10]
3	ISFs							
	ISF[11]	ISF[12]	ISF[13]	ISF[14]	ISF[15]	ISF[16]	ISF[17]	ISF[18]
4	ISFs							
	ISF[19]	ISF[20]	ISF[21]	ISF[22]	ISF[23]	ISF[24]	ISF[25]	ISF[26]
5	ISFs	Index of logarithmic frame energy						Dithering flag
	ISF[27]	g[0]	g[1]	g[2]	g[3]	g[4]	g[5]	0
6	SID Type	AMR-WB Mode Indication				Padding Bits		
	1	0	0	x	x	0	0	0
7	Padding Bits							
	0	0	0	0	0	0		

8.3.2 Frame Structure of Unvoiced Quarter-Rate

The mapping of bits for the VMR-WB Unvoiced Quarter-Rate is shown in Table 8.3-2.

Table 8.3-2: Mapping of bits in VMR-WB Unvoiced Quarter-Rate

	MSB	**Mapping of bits in VMR-WB Unvoiced Quarter-Rate**						**LSB**
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Identifier	ISFs						
	0	ISF[0]	ISF[1]	ISF[2]	ISF[3]	ISF[4]	ISF[5]	ISF[6]
2	ISFs							
	ISF[7]	ISF[8]	ISF[9]	ISF[10]	ISF[11]	ISF[12]	ISF[13]	ISF[14]
3	ISFs							
	ISF[15]	ISF[16]	ISF[17]	ISF[18]	ISF[19]	ISF[20]	ISF[21]	ISF[22]
4								
	ISF[23]	ISF[24]	ISF[25]	ISF[26]	ISF[27]	ISF[28]	ISF[29]	ISF[30]
5	ISFs	FCB Gain for Sub-Frame 1						
	ISF[31]	G[0]	G[1]	G[2]	G[3]	G[4]	G[0]	G[1]
6	FCB Gain for Sub-Frame 2			FCB Gain for Sub-Frame 3				
	G[2]	G[3]	G[4]	G[0]	G[1]	G[2]	G[3]	G[4]
7	FCB Gain for Sub-Frame 4					Unused		
	G[0]	G[1]	G[2]	G[3]	G[4]	0		

8.4 Frame Structure of CNG Eighth-Rate

The mapping of bits for the VMR-WB Unvoiced Quarter-Rate is shown in Table 8.4-1.

Table 8.4-1: Mapping of bits in VMR-WB CNG Eighth-Rate

	MSB	Mapping of bits in VMR-WB CNG Eighth-Rate						LSB
Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	ISFs							
	ISF[0]	ISF[1]	ISF[2]	ISF[3]	ISF[4]	ISF[5]	ISF[6]	ISF[7]
2	ISFs						FCB Gain	
	ISF[8]	ISF[9]	ISF[10]	ISF[11]	ISF[12]	ISF[13]	G[0]	G[1]
3	FCB Gain							
	G[2]	G[3]	G[4]	G[5]				

8.5 MIME/File Storage Format

The MIME/file storage format in VMR-WB codec simulation is activated with command line parameter "–mime". The storage format is used for storing VMR-WB encoded speech frames in a file or as an e-mail attachment.

The storage format for VMR-WB is similar to that of AMR-WB to ensure full compatibility in the AMR-WB interoperable mode. In general, VMR-WB file has the following structure:

Header
Speech Frame 1
…
…
Speech Frame n

8.5.1 Single channel Header

By default, a single channel VMR-WB file header contains only a magic number. The magic number for single channel VMR-WB files containing speech data generated in the non-interoperable modes; i.e., VMR-WB modes 0, 1, or 2, MUST consist of ASCII character string

"#!VMR-WB\n"
(or 0x2321564d522d57420a in hexadecimal).

Note, the "\n" is an important part of the magic numbers and MUST be included in the comparison; otherwise, the single channel magic number above will become indistinguishable from that of the multi-channel file defined in the next section.

The magic number for single channel VMR-WB files containing speech data generated in the interoperable mode; i.e., VMR-WB mode 3, MUST consist of ASCII character string

"#!VMR-WB_I\n"
(or 0x2321564d522d57425F490a in hexadecimal).

In the interoperable mode, a file generated by VMR-WB is decodable with AMR-WB (with the exception of different magic numbers). However, to ensure compatibility and because VMR-WB can only decode AMR-WB codec modes 0, 1, or 2, AMR-WB codec should be instructed not to generate the modes that are not in common so that files generated by AMR-WB can be decoded by VMR-WB (The compilation option EXPANDED_INTEROPERABILITY in C simulation should be selected).

8.5.2 Multi-channel Header (Currently not Implemented)

By default, the multi-channel header consists of a magic number followed by a 32-bit channel description field, giving the multi-channel header the following structure:

Magic Number
Channel Description Field

The magic number for multi-channel VMR-WB files containing speech data generated in the non-interoperable modes; i.e., VMR-WB modes 0, 1, or 2, MUST consist of the ASCII character string

"#!VMR-WB_MC1.0\n"
(or 0x2321564d522d57425F4D43312E300a in hexadecimal).

The version number in the magic numbers refers to the version of the file format.

The magic number for multi-channel VMR-WB files containing speech data generated in the interoperable mode; i.e., VMR-WB mode 3, MUST consist of the ASCII character string

"#!VMR-WB_MCI1.0\n"
(or 0x2321564d522d57425F4D4349312E300a in hexadecimal).

The 32-bit channel description field is defined as:

<table>
<tr><td colspan="8">1st Octet (MSB)</td><td colspan="8">2nd Octet</td><td colspan="8">3rd Octet</td><td colspan="8">4th Octet (LSB)</td></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="28">Reserved</td><td colspan="4">CHAN</td></tr>
</table>

Reserved bits: MUST be set to 0 when written, and a reader MUST ignore them.

CHAN (4 bit unsigned integer): Indicates the number of audio channels contained in this storage file.

8.5.3 Speech Frames

After the file header, speech frame-blocks consecutive in time are stored in the file. Each frame-block contains a number of octet-aligned speech frames equal to the number of channels, and stored in increasing order, starting with channel 1.

Each stored speech frame starts with a one-octet frame header with the following format:

<table>
<tr><td>Bit 0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>Bit 7</td></tr>
<tr><td>P</td><td colspan="4">FT</td><td>Q</td><td>P</td><td>P</td></tr>
</table>

The FT field and the Q bit are defined as follows. The P bits are padding and shall be set to 0.

Q (1 bit): Frame quality indicator. If set to 0, indicates the corresponding frame is corrupted.

Table 8.5-1: VMR-WB frame types for non-real-time transport and storage.

FT	Encoding Type	Frame Size (bits)
0	AMR-WB Interoperable Full-Rate (AMR-WB 6.6 kbps)	132
1	AMR-WB Interoperable Full-Rate (AMR-WB 8.85 kbps)	177
2	AMR-WB Interoperable Full-Rate (AMR-WB 12.65 kbps)	253
3	Full-Rate 13.3 kbps	266
4	Half-Rate 6.2 kbps	124
5	Quarter-Rate 2.7 kbps	54
6	Eighth-Rate 1.0 kbps	20
7	(Reserved)	-
8	(Reserved)	-
9	CNG (AMR-WB SID)	35
10	(Reserved)	-
11	(Reserved)	-

12	(Reserved)	-
13	(Reserved)	-
14	Erasure (AMR-WB SPEECH_LOST)	0
15	Blank (AMR-WB NO_DATA)	0

Note that in the above Table no padding for the AMR-WB compatible Frame Types is included. This is due to the fact that no frame-size adjustment for those frames is needed (to make them compatible to CDMA Multiplex Option 2), since in the file storage, no real-time over-the-air transmission takes place. Following this one octet header, the speech bits are placed as defined earlier in sections 8.1 to 8.4. The last octet of each frame is padded with zeroes, if needed, to achieve octet alignment. The following example illustrates a VMR-WB speech frame encoded at Half-Rate (with 124 speech bits) in the storage format.

<table>
<tr><th colspan="8">1st Octet</th><th colspan="8">2nd Octet</th><th colspan="8">3rd Octet</th><th colspan="8">4th Octet</th></tr>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>P</td><td colspan="4">FT=4</td><td>Q</td><td>P</td><td>P</td><td colspan="24"></td></tr>
<tr><td colspan="32" rowspan="3">Speech bits for frame-block n, channel k</td></tr>
<tr></tr>
<tr></tr>
<tr><td colspan="4"></td></tr>
</table>

Frame-blocks or speech frames that are lost in transmission and thereby not received MUST be stored as Blank/NO_DATA frames (FT=15) or Erasure/SPEECH_LOST (FT=14) in complete frame-blocks to keep synchronization with the original media (only one octet frame header is needed in this case).

9 SUPPORT FOR TDD/TTY AND LOW-RATE IN-BAND DATA

The VMR-WB codec provides support for vocoder-independent in-band data transport applications such as TDD/TTY by reserving unique bit patterns in the Full-Rate and Half-Rate encoding operation of the codec.

Since the implementation of TDD/TTY or DTMF for the cdma2000 wideband speech codec is not a requirement, the VMR-WB codec provides only the necessary support for these applications and the actual implementation of these applications is beyond the scope of this standard.

In-band data may be transported in one of three different ways:

1. Rate 1 packet with data (256 bits/packet=12800 bps)
2. Rate 1 packet with data and speech (124 bits/packet=6300 bps)
3. Rate ½ packet with data (114 bits/packet=5700 bps)

9.1 TTY/TDD Frame Format

The following unique bit patterns are reserved in VMR-WB standard for in-band data transport. Note that the reserved bit-patterns are independent of the mode of operation and are not used by the VMR-WB encoder during the processing of audio/speech signals.

9.1.1 Rate 1 with TDD/TTY Data

1111101101	256 bits of In-Band Data (TDD/TTY)

Bit Position	# Of Bits	Content
1-10	10	Preamble (Frame Identifier)
11-266	256	In-Band Data

The preamble (i.e., the first 10 bits) of the frame indicates that the frame contains in-band data (e.g., TDD/TTY) and the next contiguous 256 bits (i.e., corresponding to 12800 bits/second) contain the actual non-speech data.

However, for the purpose of satisfying the frame-level signaling requirement and thereby a need for occasional Full-Rate to Half-Rate conversion without loss of in-band data, the number of data bits should be limited to 114. For Full-Rate to Half-Rate conversion, the Full-Rate preamble bits, and the unused bits are removed by the interworking function and a Half-Rate preamble is added to the beginning of the 114 bits of in-band data to form a valid Rate-Set II Half-Rate data frame.

9.1.2 Full-Rate with TDD/TTY Data + Speech Data

1111 1010 01	126 bits of In-Band Data (TDD/TTY)	124 bits of a Half-Rate speech packet	6 unused bits

Bit Position	# Of Bits	Content
1-10	10	Preamble (Frame Identifier)
11-136	126	In-Band Data
137-260	124	Speech Data
261-266	6	Unused

The second type of Full-Rate frames with a combination of in-band and speech data uses a unique preamble as illustrated above. There are a maximum of 126 contiguous bits of in-band data (i.e., corresponding to 6300 bits/second) in addition to 124 bits of speech data (i.e., a valid Half-Rate frame) and some unused bits.

However, for the purpose of satisfying the frame-level signaling requirement and thereby a need for occasional Full-Rate to Half-Rate conversion without loss of in-band data, the number of data bits should be limited to 114 (i.e., corresponding to 5700 bits/second). For Full-Rate to Half-Rate conversion, the Full-Rate preamble bits, and the unused bits are removed by the interworking function and a Half-Rate preamble is added to the beginning of the 114 bits of in-band data to form a valid Rate-Set II Half-Rate data frame.

9.1.3 Half-Rate with TDD/TTY Data

1101 1111 01	114 bits of In-Band Data (TDD/TTY)

Bit Position	# Of Bits	Content
1-10	10	Preamble (Frame Identifier)
11-124	114	In-Band Data

The VMR-WB codec is also capable of transporting in-band data using the Half-Rate frames. The frame structure of the Half-Rate frames containing non-speech data is illustrated above. Note the unique bit pattern in the beginning of the non-speech data packet. The VMR-WB has the capability of transporting a maximum of 114 bits (i.e., corresponding to 5700 bits/second) using the Half-Rate encoding rate.